

# HEALPix を用いた「かぐや」の月面標高データベースと データ提供システムの開発

小林 寧々<sup>\*1,2</sup>, 梶浦 梨央<sup>\*1,3</sup>, 中平 聡志<sup>\*1</sup>, 山本 幸生<sup>\*1</sup>, 内田 ヘルベルト 陽仁<sup>\*1</sup>, 海老沢 研<sup>\*1</sup>

## Development of HEALPix-based KAGUYA's lunar digital elevation database and data provision system

KOBAYASHI Nene<sup>\*1,2</sup>, KAJIURA Rio<sup>\*1,3</sup>, NAKAHIRA Satoshi<sup>\*1</sup>, YAMAMOTO Yukio<sup>\*1</sup>,  
UCHIDA Herbert Akihito<sup>\*1</sup> and EBISAWA Ken<sup>\*1</sup>

### ABSTRACT

The lunar surface elevation data acquired by the KAGUYA satellite, which operated from 2007 to 2009, still has the world's highest spatial resolution for global lunar data. However, there is room for improvement in the accessibility of the KAGUYA data files archived at **DARTS**, since reading each file requires unique label information describing its content. In addition, each image is separated by equal latitude and longitude intervals, and represented with the same number of pixels; this causes a problem in precisely storing information into pixels at high latitudes. In this study, using the **HEALPix** method, which is widely used in astronomy, we divide the entire lunar globe into equal areas and define a hierarchical data structure consisting of multiple resolutions for different purposes. Thereby we successfully constructed a database that can be accessed quickly and efficiently according to users' needs. An example of a web application using this database is also presented.

### 概 要

2007年から2009年まで稼働していた「かぐや」衛星が取得した月面標高データは、全球データとしては現在でも世界最高の解像度を誇る。しかし、**DARTS**から公開しているデータはファイルごとにその内容を説明するラベル情報と照らし合わせて解読することが必要であり、アクセス性には改善の余地がある。また、緯度経度の値を等間隔に区切り、それを同じピクセル数の画像で表現しているため、高緯度での格納性が低いという問題がある。そこで本研究では、天文学で広く用いられている**HEALPix**という手法を用いて、全球を等面積に分割し、さらに用途に応じた複数解像度からなる階層的データ構造を定義することで、取得条件に適応して高速かつ簡便にアクセスできるデータセットを構築した。また、そのデータセットを用いたWebアプリケーションの例についても示す。

\* 2023年11月30日受付 (Received November 30, 2023)

<sup>\*1</sup> 宇宙科学研究所 (Institute of Space and Astronautical Science)

<sup>\*2</sup> 法政大学 (Hosei University)

<sup>\*3</sup> 鹿児島大学 (Kagoshima University)

## 1 背景

### 1.1 かぐやの月面標高数値モデル

月の代表的な探査データとして、宇宙航空研究開発機構 (JAXA) が 2007 年に打ち上げた月周回衛星かぐや (SELENE) が取得したものがあつる。月の起源と進化解明のためのデータを取得することを目的とした本衛星により、高精度な月全球の標高数値モデル (Digital Elevation Model; DEM) が作成された。月の全球 DEM は、今後人類が月探査や月科学を行つていくための基本データとして重要である。

かぐやでは前方と後方の双方を同時撮像可能な地形カメラ (Terrain Camera; TC) を搭載し、ステレオ視による DEM 生成が可能であつた。影の多い極周辺の領域はカメラによる DEM 生成が困難であるが、レーザー高度計の測距データが利用可能であり、特に極周回衛星の場合、極付近のデータ密度は高くなり DEM の高精度化に寄与する。

JAXA 宇宙科学研究所の科学データアーカイブ DARTS では、かぐやの地形カメラ TC とマルチバンドイメージャー MI、さらに米国の Lunar Reconnaissance Orbiter (LRO) に搭載されたレーザー高度計 (LOLA) のデータを使用して作成された月の全球 DEM 「SLDEM2013」 [1] を公開している。その後、米国の GRAIL 衛星により月重力場モデルの大幅な改善に伴い LOLA データも改善され、それをういた SLDEM2015 [2] が  $\pm 60$  度の低緯度から中緯度までの緯度帯のみで作成されたが、全球 DEM は公開されていない。全球 DEM としては、現在でも SLDEM2013 が世界で最も高精度である。

### 1.2 DARTS における DEM データの保管と課題

前項で説明した SLDEM2013 データは、全月球を緯度経度方向に  $1^\circ \times 1^\circ$ 、4096 pixel  $\times$  4096 pixel で区切って収録した、PDS3 (Planetary Data System ver.3) 形式のデータである。この貴重なデータの活用を促進するために、Web アプリケーション等を通して、全月球のあらゆる位置に対して、簡便かつ迅速に高度データが得られるようにすることが望ましい。しかし、それには、以下に示す二つの課題が存在する。

一つ目はデータ格納性の課題である。現在のデータ形式では月面標高データを緯度経度空間で一定間隔で分割して FITS ファイルとして格納するため、月面の場所によって一つの FITS ファイルに含まれる面積が異なる。例えば、最も面積の大きい赤道と緯度 60 度を比較すると、1 ピクセルの幅が  $1/2(\cos 60^\circ)$  に変化するので、面積は 4 倍変わることになる。ファイルごとに格納面積が異なるにも関わらず 1 つの FITS ファイルに含まれる画素数は全て 4096 pixel  $\times$  4096 pixel で統一されているため、高緯度地域では 1 ピクセルあたりの面積がさらに小さくなり、見かけ上の解像度が観測性能を大きく超えてしまう。このように場所ごとに解像度の違いがあることで、データの格納効率が低下する。

二つ目は、データの利用性の問題である。PDS 形式のデータは対象のファイルごとにその内容を記述するラベル情報を照らし合わせて解釈する必要がある。現在 DARTS で保管されている PDS3 形式のデータは緯度経度方向に  $1^\circ \times 1^\circ$  のエリアの高度情報を FITS ファイルで記録し、各 FITS ファイルが記録するエリアの情報を 1 つのラベルファイルによって説明している。そのため、例えば、ある地点周辺何 km と言つた特定の月面領域データを取得したい場合、その領域をカバーする複数のファイルについて、各ファイルの境界を考慮に入れてラベルファイルを読み解く必要がある。また、一つ目の問題点で説明したように、緯度による面積の変化を解決するために、複雑な処理が必要になってしまう。

本開発ではこれら二つの課題を解決し、月面上の任意の入力位置に対して効率的なデータ取得が可能な月面標高データベースを構築する。それによって、アクセス性の高い Web アプリケーションの実現を目指す。

## 2 HEALPix を用いた月面標高データ格納方式の開発

### 2.1 HEALPix を利用することによる利点

本研究では、HEALPix (Hierarchical Equal Area isoLatitude Pixelation) と呼ばれるピクセル化アルゴリズム [3] を用いたデータ格納方式を採

用した。HEALPix は天文学において全天データを扱うために開発されたデータ形式であり、全球を等面積のタイルに分割するものである。HEALPix にはピクセル順序の方式 (ordering) として “NEST”, “RING” の二種類があり、分割数を決定するためのパラメータとして  $N_{\text{side}}$  が与えられ、 $N_{\text{side}}$  には通常 2 のべき乗数を用いられる。HEALPix を用いると全球のデータは、 $12 \times N_{\text{side}}^2$  の長さの一次元配列 (hpx) として表現でき、 $i$  番目のピクセルに格納された値は  $\text{hpx}[i]$  として参照可能である。このような性質を踏まえて、我々は HEALPix を利用した月面標高データの格納方法を開発することで以下のような利点が得られると考えた：

1. 全月球が等面積に分割されるので、標高データを緯度経度によらず全て同じ解像度で格納することができる。
2. ordering と  $N_{\text{side}}$  を決定すると、緯度経度に対するピクセル番号が HEALPix の定義から求まり、データの格納位置が決まる。
3. 更に、ordering として “NEST” を用いた時に  $N_{\text{side}}$  を  $N_{\text{side}}^a$  から  $N_{\text{side}}^b$  ( $< N_{\text{side}}^a$ ) に変換する場合を考える。これは、画像の解像度を変更することに対応する。このとき、 $N_{\text{side}}^a$  における任意のピクセル  $i_a$  は、 $N_{\text{side}}^b$  におけるピクセル

$$i_b = \text{int} \left( i_a \times \left( \frac{N_{\text{side}}^b}{N_{\text{side}}^a} \right)^2 \right)$$

の中に含まれることが、HEALPix の定義から決定される (図 1)。

4. 更に、ピクセル番号  $i_b$  に内包される  $N_{\text{side}}^a$  のピクセル番号は、 $i_b \times \left( \frac{N_{\text{side}}^a}{N_{\text{side}}^b} \right)^2$  から  $(i_b + 1) \times \left( \frac{N_{\text{side}}^a}{N_{\text{side}}^b} \right)^2 - 1$  の間で連続する。

このような HEALPix の性質を活かすことで、階層的なデータベースの構築やデータへのアクセスが容易になり、ローカルでの解析用データベースだけでなく、Web アプリケーションなど、利用者の要求に応じて即時にレスポンスが求められる用途にも HEALPix が有用となる。

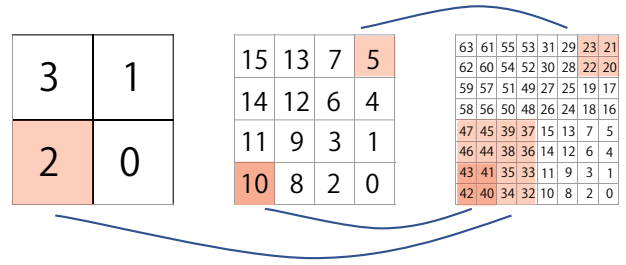


図 1 HEALPix における解像度変換の模式図 (“NEST” を用いた場合)

## 2.2 HEALPix を用いたデータ格納方式の構築

前項では、HEALPix を用いることで全月球の高度データを原理的に 1 つの配列で表現できることを示したが、全球をカバーする「かぐや」DEM データの容量は約 2 TB に及ぶ。よって、それに対応する HEALPix 配列をメモリ上、またはファイル上に格納することは現実的ではない。そのため、適切な粒度で HEALPix を分割することを検討した。ordering として以降では、前項で説明したように解像度の変換を容易に行うことが可能な “NEST” を用いる。

今回作成する HEALPix の解像度は、元データの解像度を失わないようにするため、当初  $N_{\text{side}} = 2^{18}$  を検討した。しかしこの場合、赤道付近ではオーバーサンプリングになり、補完の処理が煩雑になる。また、高度の値 (メートル) に元データと同じ 16bit 整数を用いるとデータ量が約 1.5 TB となる。これでも Web アプリケーション用データとして配置するには現時点では大きすぎるので、1 段階下げた  $N_{\text{side}} = 2^{17}$  を今回作成する最大解像度のデータと位置づけた\*\*1。 $N_{\text{side}} = 2^{17}$  を用いることで、全月球は約 2 千億個のピクセルに分割され、この場合 1 ピクセルの長辺の長さは約 13.5 m となる。

HEALPix ピクセルとファイルを分割する粒度は、実際の用途における解像度ごとの利用ケースを想定して、(1) 1 つのファイルに含まれる領域が広すぎず、(2) 必要な範囲のデータを得るために開くファイル数が多すぎない、という 2 つの要求バランスから、全月球を  $N_{\text{side}}^P = 2^{17}$  でピクセルに分割した際

\*\*1 将来的に、必要に応じて  $N_{\text{side}} = 2^{18}$  のデータベース作成も検討する。

に、 $N_{\text{side}}^{\text{F}} = 2^7$  の分割数のタイルごとに 1 ファイルを作成するように設定した。この時、 $N_{\text{side}}^{\text{F}}$  で定義された長辺は人間の目線から見た地球の水平線までの距離の約 5km よりを参考に、やや大きい約 14km とした。そうすることで、約 20 万ファイルが作られるが、これを並列に配置するとリスト表示に時間がかかるなど、ファイルの扱いとして不便になるので、 $N_{\text{side}}^{\text{D}} = 2^3$  に相当するタイルごとにファイルを入れた（つまり 768 ディレクトリに 256 ファイルずつが格納されている）。

解像度が高いデータは、人間のスケールで月に降り立った状態の表示など、近接して表示する際に用いられ、解像度の低いデータは引いた画角で、より広い範囲を表示するために用いられる。そのため、1 ファイルあたりに含めるピクセル数および、1 ディレクトリに格納するファイル数は維持しながら、表 1 に示すように最も細かい解像度を  $N_{\text{side}} = 2^{17}$  とし、それ以外に、 $2^{16}$ ,  $2^{15}$ ,  $2^{14}$  の異なる解像度の階層を設定した。

$N_{\text{side}}^{\text{P}}$ (ピクセルサイズ)	$N_{\text{side}}^{\text{F}}$ (範囲)	$N_{\text{side}}^{\text{D}}$
$2^{17}$ (~14 m)	$2^7$ (~14 km)	$2^3$
$2^{16}$ (~27 m)	$2^6$ (~28 km)	$2^2$
$2^{15}$ (~54 m)	$2^5$ (~56 km)	$2^1$
$2^{14}$ (~108 m)	$2^4$ (~111 km)	$2^0$

表 1 作成した階層的なデータ構造。  $N_{\text{side}}^{\text{P}}$  は、ピクセルの分割単位 (1 ピクセルのサイズ)、 $N_{\text{side}}^{\text{F}}$  はファイルのタイルの分割単位 (1 ファイルに含まれるピクセル群が覆う範囲)、 $N_{\text{side}}^{\text{D}}$  はディレクトリの分割単位を表す。

### 2.3 具体的なデータ配置とデータの抽出

前項までに説明したような方針に従って、実際のファイルとディレクトリの構造を構築した。この場合 1 つのディレクトリあたりには、

$$L = \left( \frac{N_{\text{side}}^{\text{F}}}{N_{\text{side}}^{\text{D}}} \right)^2$$

個のファイルの、ファイル番号  $i_{\text{F}}$  が、

$$i_{\text{F}} = L \times i_{\text{D}} + (0, 1, 2, \dots, L - 1)$$

の範囲のものが格納されることになる。また、1 つのファイルあたりには

$$M = \left( \frac{N_{\text{side}}^{\text{P}}}{N_{\text{side}}^{\text{F}}} \right)^2$$

個のピクセル番号  $i_{\text{P}}$  が以下のような順序で格納されている。

$$i_{\text{P}} = M \times i_{\text{F}} + (0, 1, 2, \dots, M - 1)$$

このような規則で決定されるファイル及びディレクトリ名を、

$$\text{Dir}[i_{\text{D}}]/\text{sldemhpx-}[i_{\text{F}}].\text{fits}$$

とした。各ファイルは、HEALPix のパラメータなどを書き込んだヘッダ情報を含む FITS 形式で作成した。図 2 にはより具体的なファイルのツリー構造の例を示した。すべてのピクセルのピクセル番号を表すには 64 bit が必要であるが、高度は 16bit 整数で格納されているので、64 bit のインデックス情報を伴うことになるとデータサイズが大幅に増加してしまう。しかしすでに説明したように、ファイル名とファイル中の配列の順番から、HEALPix のピクセル番号が自明に決められるため、作成するファイルではピクセル番号は省略し、高度のみが書き込まれた一次元配列となっている。

$N_{\text{side}}^{\text{F}} = 2^{17}$  では 1 つの HEALPix のピクセル値は元 DEM 画像の 1 から数ピクセルの平均となるので、元 DEM データと同じ符号付き 16 bit 整数で、ダイナミックレンジを活用するために 3 倍して格納し、合計で約 390GB のデータとなった。 $N_{\text{side}}^{\text{F}} = 2^{16}$  以下ではより多くの元ピクセルの平均値となるため、32 bit 浮動小数点データとして格納した ( $N_{\text{side}}^{\text{F}} = 2^{16}$  の場合約 190GB、以降は解像度が下がる毎に 4 分の 1)。

作成したデータから特定の緯度経度の値を取り出したい場合は、 $N_{\text{side}}^{\text{P}}$  を決定すると、緯度経度に対応するピクセル番号  $i_{\text{P}}$  が HEALPix の定義から計算でき、 $i_{\text{P}}$  の格納位置は、ファイル番号  $i_{\text{F}} = \text{int}(i_{\text{P}}/M)$  の ( $i_{\text{P}}/M$ ) 番目であり、そのファイルはディレクトリ  $i_{\text{D}} = \text{int}(i_{\text{F}}/L)$  の中に存在することがわかる。



着目したエリアの画像や 3D メッシュを作成する利用ケースでは、ある領域のデータをまとめて取り出すことになる。その場合は、着目する領域に含まれるファイルを、 $N_{\text{side}}^F$  から計算し、ファイル名に対応するピクセル番号からそのエリアのタイル (ファイル) を読み出し、データの格納規則から高度データに HEALPix ピクセル番号を割り当て、その後の処理を行う。

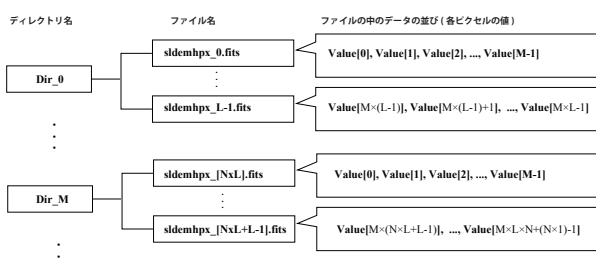


図 2 ディレクトリ構造

## 2.4 全球 DEM データの HEALPix 化変換処理の実装

DARTS から公開されている、緯度経度で分割された DEM 画像データ 64800 枚、合計約 2TB のデータセットを §2.1-2.3 で説明した HEALPix で定義されたファイルフォーマットに変換する処理を行った。変換処理の過程ではデータ量が膨大なので単純な処理では現実的な時間で終わることが難しかったが、以下のように処理することで、処理全体を半日以内で終わることができた。

### 2.4.1 1枚あたりの処理の高速化・効率化のための処理方法

まず画像 1 枚 1 枚の処理では、以下のような手順で処理を行なった。

1. 元 DEM データの画像データを 1 対 1 で HEALPix の高度データに変換した。各処理では、画像内の各ピクセルの緯度経度を HEALPix ピクセル番号に変換して、ピクセル番号をインデックスにした高度の値を取り出し、同じピクセル番号が複数存在する場合はそれらの平均を取った。
2. 最も細かい解像度に対応する  $N_{\text{side}}$  の、ピクセル

番号付きの高度データを §2.2 で説明した構造に変換した。

3. 変換後のデータを元に、解像度を 1 段階ずつ下げる処理を順次行なった。

処理の過程では、広い値域に散らばって出現するピクセル番号と対応する高度データの集計処理を行う必要がある。その高速な集計にはヒストグラムの処理が適しているがその場合、全地球のピクセル番号に近いサイズの配列をメモリ上に確保しなければならない。ただし、出現するピクセル番号の種類は、全地球のピクセル数に対して数万分の 1 程度と「疎」なので、疎行列 (sparse matrix) を用いて処理することにした。疎行列の処理には具体的には、TensorFlow[4] の疎行列パッケージの `bincount()` 関数を用いることで少ないメモリで処理を実行でき、処理全体としても簡素化および高速化が実現できた。

### 2.4.2 並列処理による高速化

1 枚あたりの処理を効率化した上でも元データの枚数が膨大なので、JAXA のスーパーコンピュータ (JSS3<sup>\*\*2</sup>; JAXA Supercomputer System generation 3) を用いた。独立な変換処理を並列で行うことで高速化した「ジョブ投入最適化ツール」を用いて 1 プロセスあたり 1CPU、同時実行上限数を約 500 プロセスに設定し、上記のようにメモリ消費量を削減した事で、1 プロセスあたりの割当メモリ量は 2 GB で問題なく処理を実施できた。

## 3 HEALPix を用いた「かぐや DEM」データベースを利用した Web アプリケーションの実装

### 3.1 Web アプリケーションとしての実装

DARTS では Web アプリケーション提供の環境としてオンプレミスからクラウドへの移行を徐々に進めており、本開発においてもクラウドサービス、具体的には Microsoft 社の Azure を利用してアプリ

<sup>\*\*2</sup> <https://www.jss.jaxa.jp>

ケーションを構築した<sup>\*\*3</sup>。本開発では、最もシンプルなアプリケーションとして、月の緯度経度と抽出するエリアを表す半径の値を入力して、画像と 3D のポリゴンメッシュデータを作成する機能だけを含めることにした。

開発した Web アプリケーション全体の構成図は図 3 に示す。フロントエンドは Node.js のフレームワークである Vue.js を用い「Azure Static Web Apps」に配置した。バックエンドは Python を使って「Azure Functions」アプリとして実装した。データベースの格納および出力の書き込み先としては、大容量のデータに向く「Azure Blob Storage」を用いた。ソースコードは GitHub に格納し CI/CD (Continuous Integration/Continuous Delivery) を用いた開発とリリースを実装している。このアプリケーション処理の流れは以下のようになっている：

1. フロントエンドに Web ブラウザーでアクセスし、着目する緯度経度と抽出領域の半径を入力し、実行ボタンを押す。
2. フロントエンドアプリが Azure Functions アプリを起動する。
3. Azure Functions アプリが、§2 で説明したファイル構造に従って Blob Storage に格納されたデータを読み取り、画像データとポリゴンメッシュデータを生成する処理を実行して、出力用 Blob Storage に結果ファイルを書き込む。
4. 書き込んだ Blob Storage のパスに対応する URL をフロントエンド経由でユーザーに返す。
5. 画像はそのままブラウザに表示され、ポリゴンメッシュデータは同時に表示される URL にアクセスすることで取得可能である。

現在の実装では、処理しきれないデータ処理要求を避けるため、利用するデータベースの解像度は要求する半径によって自動的に決定する事とし ( $\sim 0.2$  度で  $N_{\text{side}}=2^{17}$ )、最大半径は  $10^\circ$  ( $\sim 300$  km) とした。処理にかかる時間は入力に応じて変化するが、

<sup>\*\*3</sup> こちらの URL から開発版が利用可能である。  
<https://calm-bush-0abfea300.1.azurestaticapps.net/>

長くても 20 秒強から 30 秒程度で処理を終える事が確認できた。出力された画像と、レイトレーシングによって可視化したポリゴンメッシュデータの例を図 4 に示した。

### 3.2 画像データの出力について

全地球のデータを画像ファイルとして作り置きし、素早く表示するアプリはこれまでに DARTS 内でも別の実装があり、表示速度の面では本アプリより早く動作する可能性がある。しかし、全球を予め画像化し、人工的な切れ目なく表示するためには、グローバルにカラーマップを決める必要がある。いっぽう、今回開発したアプリでは着目した領域に対して適応してデータを処理しており、データの表示フレーム内の情報からローカルにコントラスト等を決める事で、細かい構造をわかりやすく可視化することが可能となっている。

### 3.3 HEALPix 点群データのメッシュ化

本研究で構築した HEALPix ベースの情報を、ゲームや 3DCG その他の用途に利用な三次元データに変換するためのアルゴリズムについて説明する。HEALPix ベースの DEM データから、任意の緯度経度に対する高度の値、更には三次元空間上の点群データを取得することは容易である。ただしこの状態の点群データだけでは面または法線を表現できず、3D データとしては不完全である。そのため、3D データとして扱いやすいポリゴンを使ったデータフォーマットでメッシュデータを作成することにした。HEALPix の定義から、図 5 の様にピクセルの並び、着目したピクセルと周辺の関係は自動的に決まっている。今回の実装ではデータに含まれる各ピクセルの境界を構成する 4 点の座標を参照し、それらのうち 3 つを結ぶ三角形を 2 つ生成した。これを繰り返すことで、HEALPix の定義に基づく並び方をした点群データをメッシュ化することができた。出力フォーマットは最も多くの 3D 関連ソフトウェアでサポートされている STL (Standard Triangle Language, Standard Tessellation Language) を採用した。例えば、月の緯度経度の座標から半径  $0.5^\circ$  以内の範囲をメッシュ化する場合、解像度  $N_{\text{side}} = 2^{17}$  の場合に

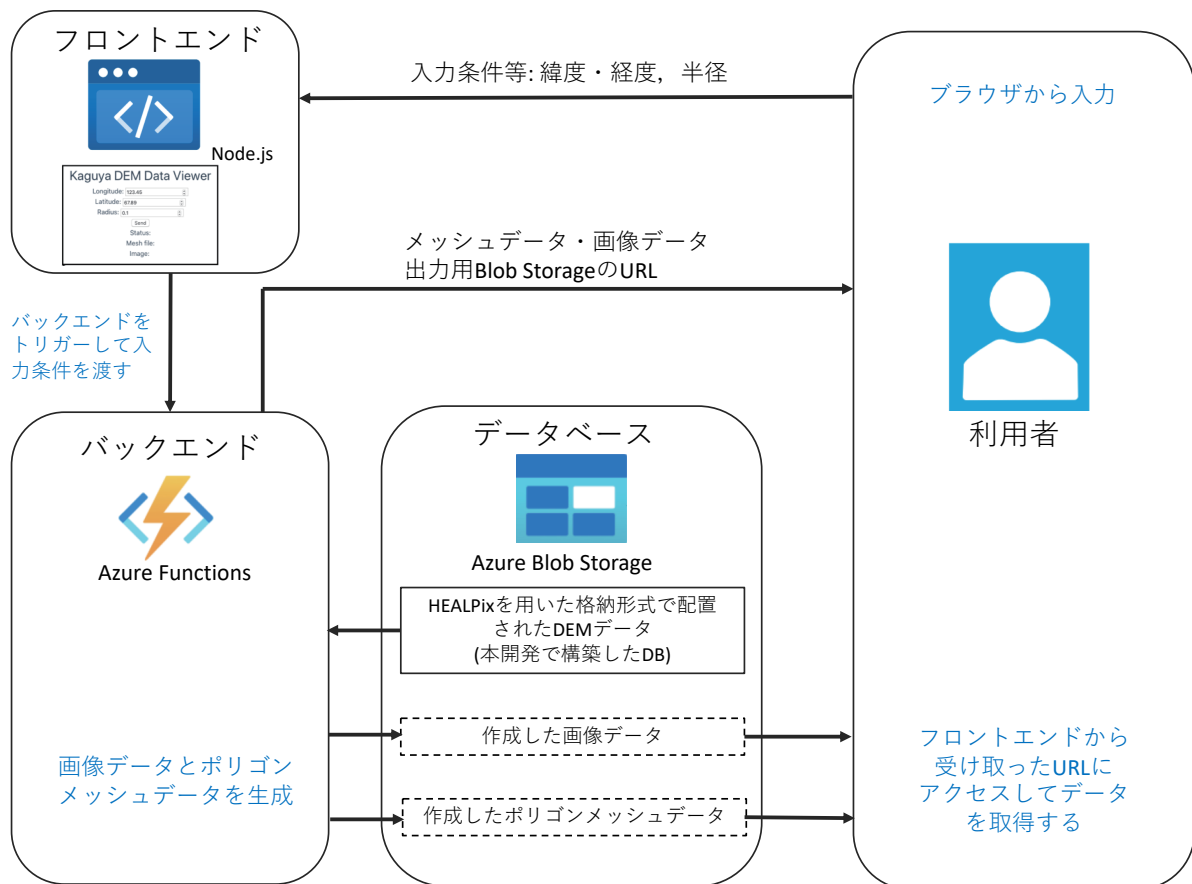


図3 Azure上に構築したWebアプリの構成

かる処理時間は、ローカル PC 環境で、10 秒程度 (データサイズは 350MB) となった。この処理時間は Web アプリでの利用でも十分に実用的な範囲だと考えられる。

#### 4 まとめと今後の展望

DARTS から公開されているかぐや DEM データ SLDEM2013 を元に、天文学でよく使われる HEALPix を用いたデータベースを作成した。また、そのデータベースを用いて、両極を含む任意の月のエリアに対する月面高度マップ画像や、3DCG 向けのポリゴンメッシュデータをリアルタイムで提供する Web アプリケーションを構築し、試験公開を開始した。

今回構築したデータの階層化を用いたデータアクセスの効率化手法と比較可能な手段としては、タイル化した画像ファイルを用いる方法が挙げられ、

HEALPixベースで実装された HiPS<sup>\*\*4</sup>等も広く使われている。今回用いた方法はそれと比較すると、データから画像を生成する処理が入る分表示の即応性は落ちるが、生成する見やすさを最適化させられる点 (§3.1)、3D メッシュを様々な解像度で作りやすい (§3.2) という点でメリットがある。

現在このデータベース化の対象としているのは DEM データだけで、月面の光学的な特性を表したマップは未処理であるため、CG として利用した際の見栄えは不十分である。そのためのデータ処理は今後の課題である。また、作成したアプリケーションはデータを提供するだけの最もシンプルなものとなっている。このデータセットをフルに活かしたアプリケーションとして、全球データがシームレスに階層化されているという特徴を用いて、月面の走

<sup>\*\*4</sup> <https://aladin.cds.unistra.fr/hips/>

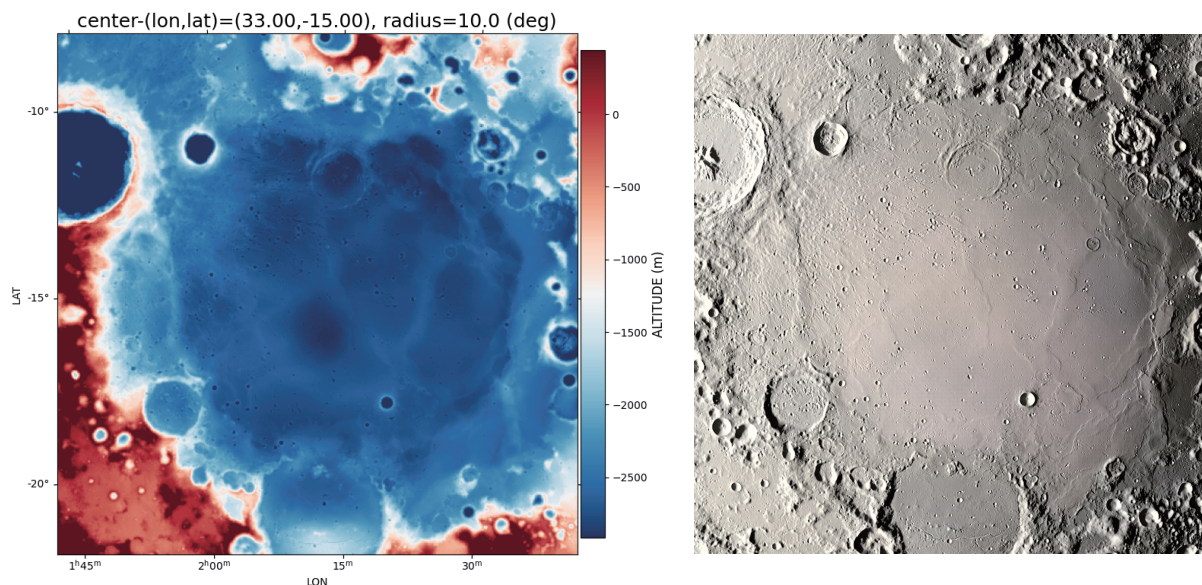


図4 月面の東経33度, 南緯15度周辺の半径約10度付近における(左)月面高度データのカラーマップ画像と(右)3Dメッシュ化したデータに平行光線を照射して得たレイトレーシング画像。

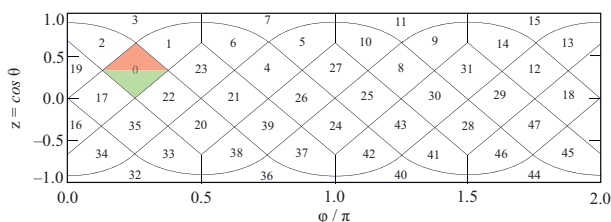


図5 HEALPix点群データをメッシュ化する際の定義. それぞれのHEALPixピクセルが三角形のポリゴンで2分割(赤, 緑)される.

行・飛行に従って動的に3Dデータを更新しながら表示するようなアプリケーションを作りたいと考えている。

## 5 謝辞

本研究では, 宇宙航空研究開発機構・宇宙科学研究所が運用する科学データアーカイブDARTSから公開されているデータを用いた。また, 宇宙航空研究開発機構スーパーコンピュータシステムJSS3を, 「JSS3を用いた科学衛星データ処理の検討」の事業名(事業コード: EU30800)で利用した。HEALPixを用いたデータ処理では, healpyライブラリ[5]を利用した。

## 参考文献

- [1] J. Haruyama, M. Ohtake, T. Matsunaga, H. Otake, Y. Ishihara, K. Masuda, Y. Yokota, and S. Yamamoto. Data Products of SELENE (Kaguya) Terrain Camera for Future Lunar Missions. In *45th Annual Lunar and Planetary Science Conference*, Lunar and Planetary Science Conference, p. 1304, March 2014.
- [2] M.K. Barker, E. Mazarico, G.A. Neumann, M.T. Zuber, J. Haruyama, and D.E. Smith. A new lunar digital elevation model from the lunar orbiter laser altimeter and selene terrain camera. *Icarus*, Vol. 273, pp. 346–355, 2016.
- [3] Krzysztof M. Gorski, Benjamin D. Wandelt, Frode K. Hansen, Eric Hivon, and Anthony J. Banday. The HEALPix Primer. *arXiv e-prints*, pp. astro-ph/9905275, May 1999.
- [4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving,



Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

- [5] Andrea Zonca, Leo Singer, Daniel Lenz, Martin Reinecke, Cyrille Rosset, Eric Hivon, and Krzysztof Gorski. healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in python. *Journal of Open Source Software*, Vol. 4, No. 35, p. 1298, March 2019.