

HEALpixを用いた「かぐや」の月面標高データへの アクセス性向上とデータ提供システムの検討

小林寧々(法政大), 梶浦梨央(鹿児島大),
中平聡志, 山本幸生, 海老沢研 (ISAS/JAXA)

目次

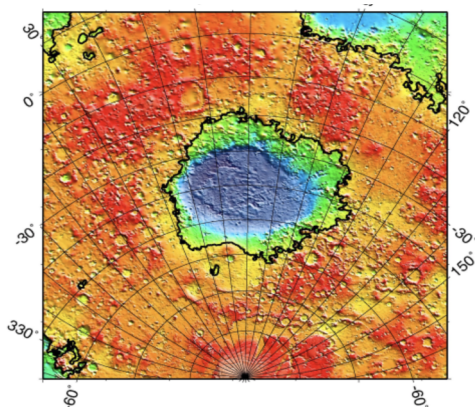
- 背景
- 目的
- 手法
 - DEMデータのHealPix化
 - 階層構造の決定とフォルダ構築
 - データ提供システムの初期検討
- 今後の展望

月面標高データとは

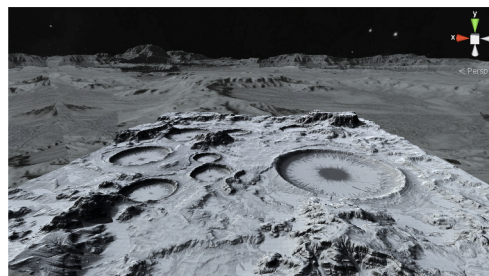
- 標高データ
 - 地形をメッシュ上に区切り、各点の標高値を記載したもの
- DEM(Digital Elevation Model)
 - 建物や樹木などを取り除いた地表面の高さ表示

今後研究目的に限らず広範な用途での利用が見込まれる

- (例) ビジネス目的（ゲームやアプリ）など



使用例1 地形解析



使用例2 ゲーム製作

DARTSから公開されているかぐやDEMデータ

- JAXA宇宙科学研究所の宇宙科学データアーカイブ
DARTS で公開されている
 - <https://data.darts.isas.jaxa.jp/pub/pds3/sln-l-tc-5-sldem2013-v1.0/>

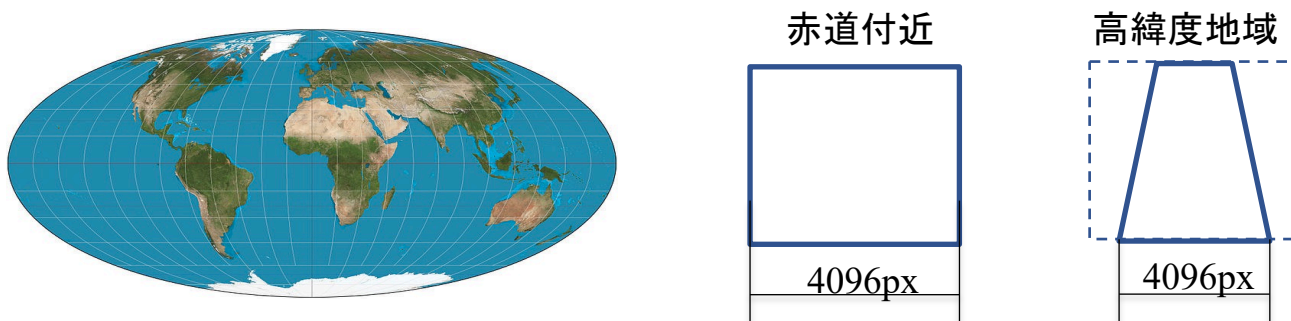
Index of /pub/pds3/sln-l-tc-5-sldem2013-v1.0/ion000/data

Name	Last modified	Size	Description
Parent Directory	-	-	-
DTM_MAP_01_N00E000S01E001SC.img	2016-01-14 21:09	32M	
DTM_MAP_01_N00E000S01E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N01E000N00E001SC.img	2016-01-14 11:59	32M	
DTM_MAP_01_N01E000N00E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N02E000N01E001SC.img	2016-01-14 19:34	32M	
DTM_MAP_01_N02E000N01E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N03E000N02E001SC.img	2016-01-14 14:39	32M	
DTM_MAP_01_N03E000N02E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N04E000N03E001SC.img	2016-01-25 11:44	32M	
DTM_MAP_01_N04E000N03E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N05E000N04E001SC.img	2016-01-14 14:57	32M	
DTM_MAP_01_N05E000N04E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N06E000N05E001SC.img	2016-01-14 22:02	32M	
DTM_MAP_01_N06E000N05E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N07E000N06E001SC.img	2016-01-25 12:26	32M	
DTM_MAP_01_N07E000N06E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N08E000N07E001SC.img	2016-01-14 13:34	32M	
DTM_MAP_01_N08E000N07E001SC.lbi	2017-03-21 15:35	3.3K	
DTM_MAP_01_N09E000N08E001SC.img	2016-01-14 19:01	32M	
DTM_MAP_01_N09E000N08E001SC.lbi	2017-03-21 15:35	3.3K	

- ・ かぐやのTerrain Camera (TC)から得られたデータ
- ・ 緯度経度1°で1ファイルとしてデータが公開されている

かぐやDEMデータの課題

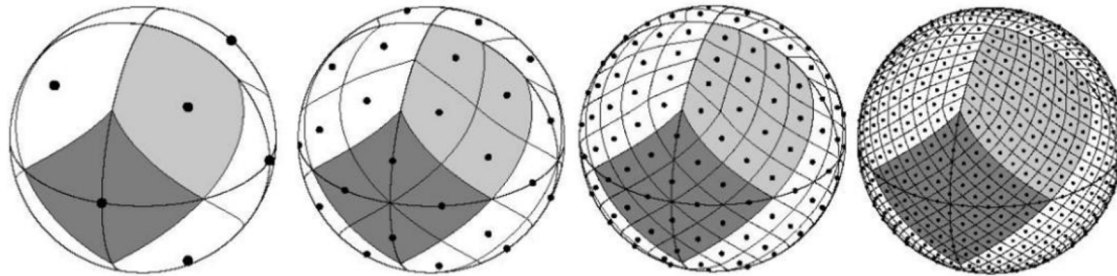
- ファイルごとにラベル情報を照らし合わせて解釈する必要がある
 - 広範な用途で利用に対応するには任意の条件で簡単に抽出できる必要があり、その際、即時性も重要である
- データ・フォーマットの特性上、場所によって解像度が異なる
 - 赤道と緯度60度を比較すると、ピクセルの幅が2倍($\cos 60^\circ$)、面積が4倍変わる。高緯度では更に面積の差が出てくる



上記の理由からデータ提供時にデータの解釈や解像度変換などいくつかの処理が必要になり即時のデータ提供が難しい提供しやすいデータベースの構築が必要と考え、それら解決に取り組んだ

HEALPix を用いた月地形データベース

- HEALPix : **H**ierarchical **E**qual **A**rea iso**L**atitude **P**ixelation
 - 天文学の全天データを扱うために開発されたデータ形式
 - 球を緯度方向に一定間隔になるよう分割した等面積のピクセル
 - 分割数(Nside)を決めると、着目した緯度経度に対応するピクセル番号が一意に求まる



これらを利用してHEALPixベースのデータベースを構築できないかと考えた
データが大きい(元のDEM画像が2TB)ためメモリ上で1つの配列に収まらない
→ うまく構造化、階層化する必要がある。

目的と目標

目的

- かぐやの大規模データを簡単に利用できるようデータフォーマットを変換すること

目標

- HEALPixを用いた月面標高データベースを構築
- 扱いやすいデータサイズでの階層化
- アクセスの容易なデータ提供システムの構築

手法

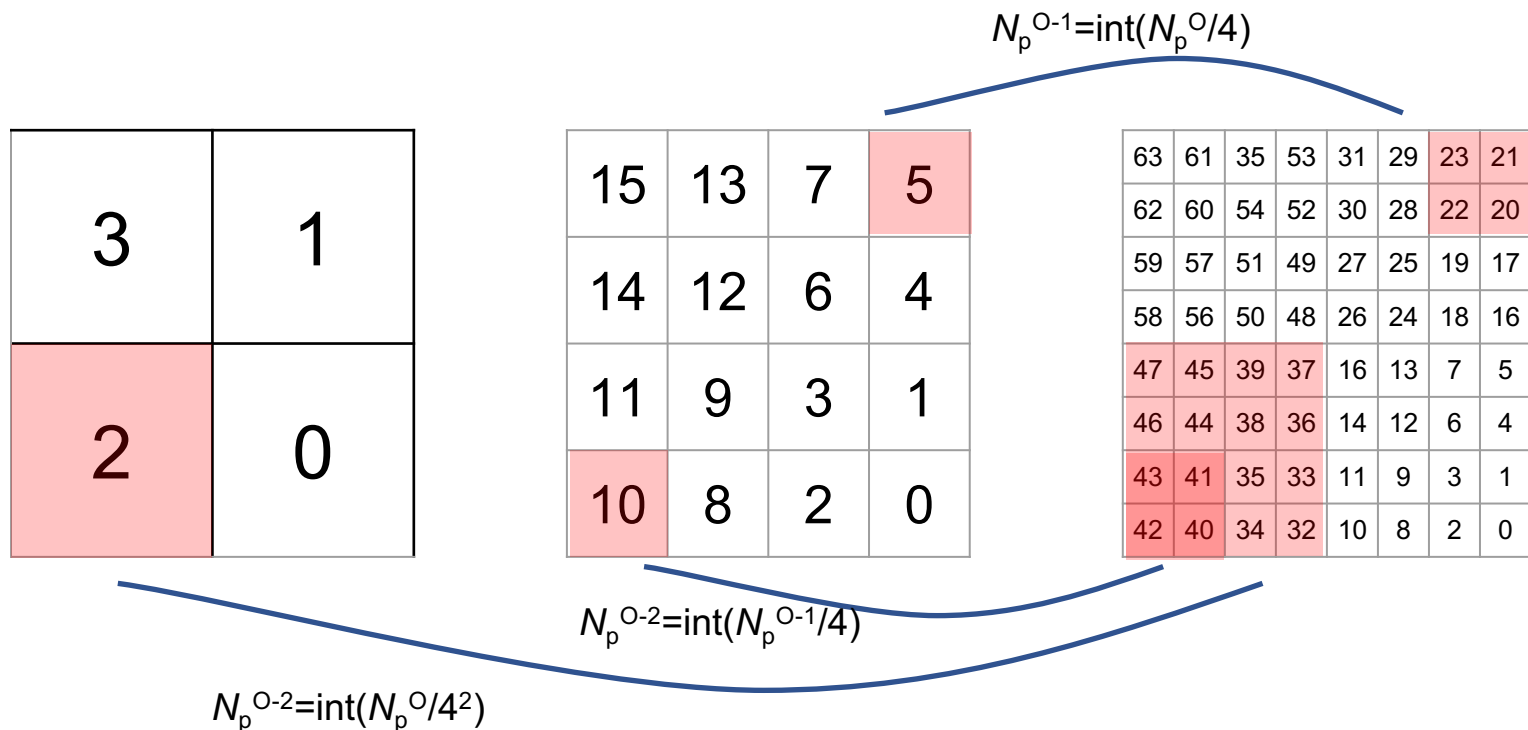
1. DEMデータのHealPix構造化
2. データ処理の高速化
3. データ提供システムの初期検討

HEALPixをもとに階層構造を構築する必要性

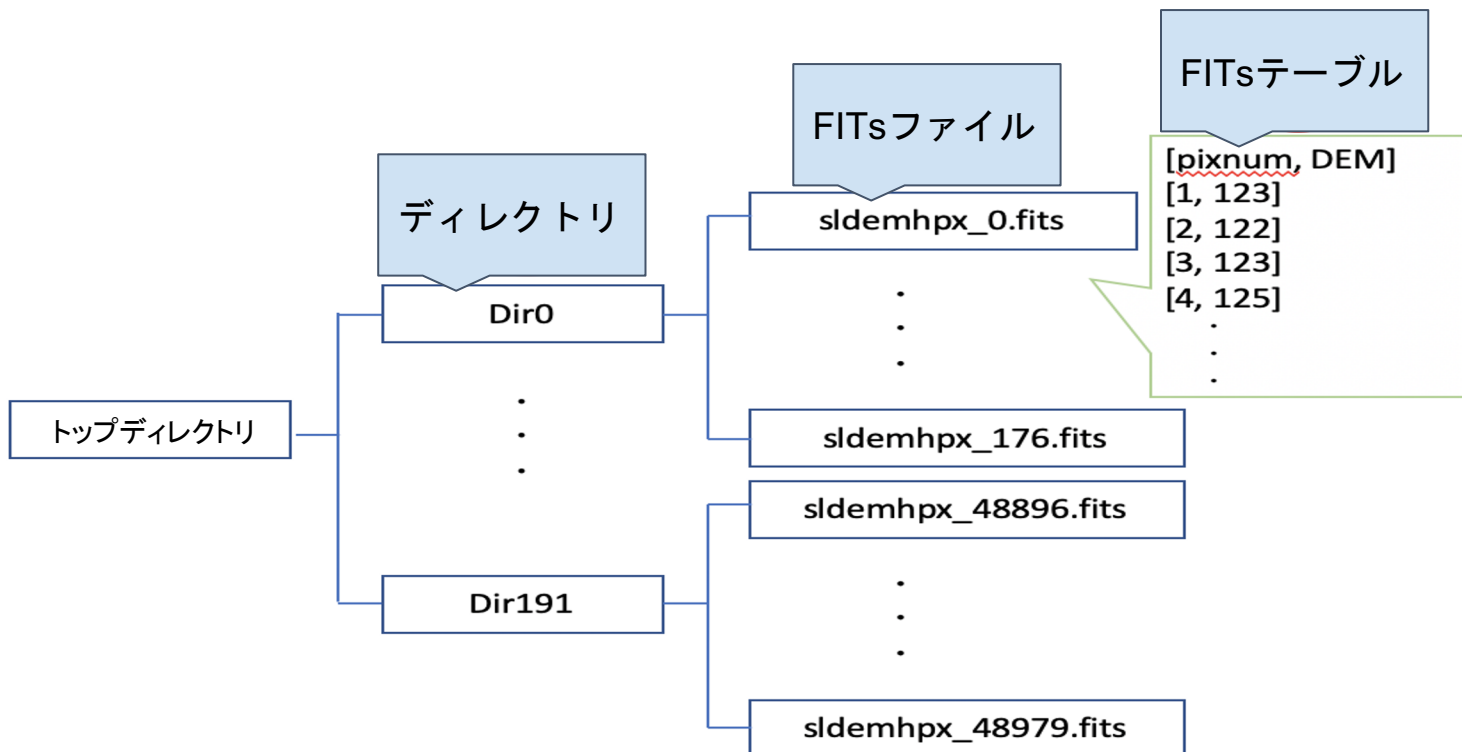
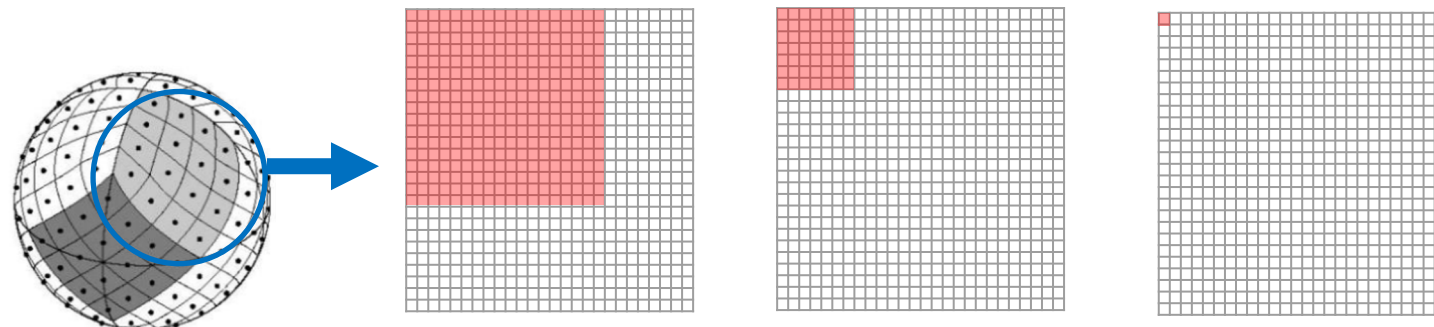
- HEALPix化すれば全球の値を1つの配列で表現できる。
 - あるLON,LATに対応するHEALPixピクセル番号は定義と指定したNside(分割数)から定まる
 - あるピクセル番号に対応する数値(高度等)がどこに格納されているかも自明にわかる
- 「かぐや」DEMデータの場合はデータ量が膨大なので1つのファイルにデータ全体を格納することは難しい
 - そのため、配列を使いやすい形に分割する

HEALPixの性質

- HEALPixのピクセル定義 (NEST) を使用することで、ピクセル番号をそれをより粗いNside(分割数)でのピクセル番号へ簡単な計算で変換可能



目指した階層構造



データ検索の実例

前提条件

データとしての分割単位:

$N_{order_F}=17 \rightarrow N_{side}=131072$ 分割数~206億ピクセル

ファイルの分割単位:

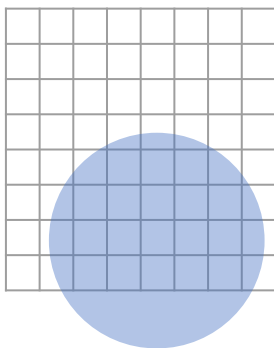
$N_{order_M}=7 \rightarrow N_{side}=128$ 分割数~19.7万ファイル

ディレクトリの分割単位:

$N_{order_R}=3 \rightarrow N_{side}=8$ 分割数~768ディレクトリ

入力[例]

(LAT, LON)=(67.89, 123.45)の高度の値を取り出したい



値を取り出す際の計算

(LAT, LON)に対応するピクセル番号は32688887481 (HEALPixの定義より)

ファイルの番号は $\text{int}(32688887481 \times (2^{N_{order_M}})^2 / (2^{N_{order_F}})^2) = 31174$

ディレクトリの番号は $\text{int}(32688887481 \times (2^{N_{order_R}})^2 / (2^{N_{order_F}})^2) = 121$

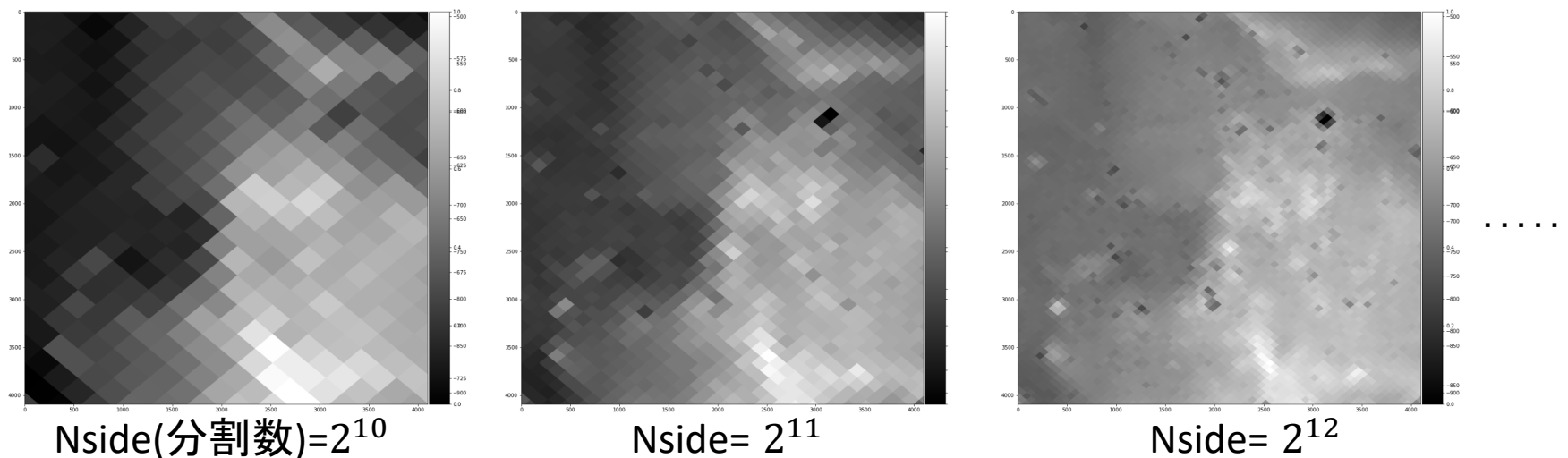
→読むべきファイルは Dir121/file_31174.fits

ファイル先頭データのピクセル番号は、 $31174 \times (2^{N_{order_F}})^2 / (2^{N_{order_M}})^2 = 32688308224$

$32688887481 - 32688308224 =$ ファイルの中の配列の579257番目に目的の値がある

データとしてのHEALPix解像度

● NsideごとのHEALpix比較

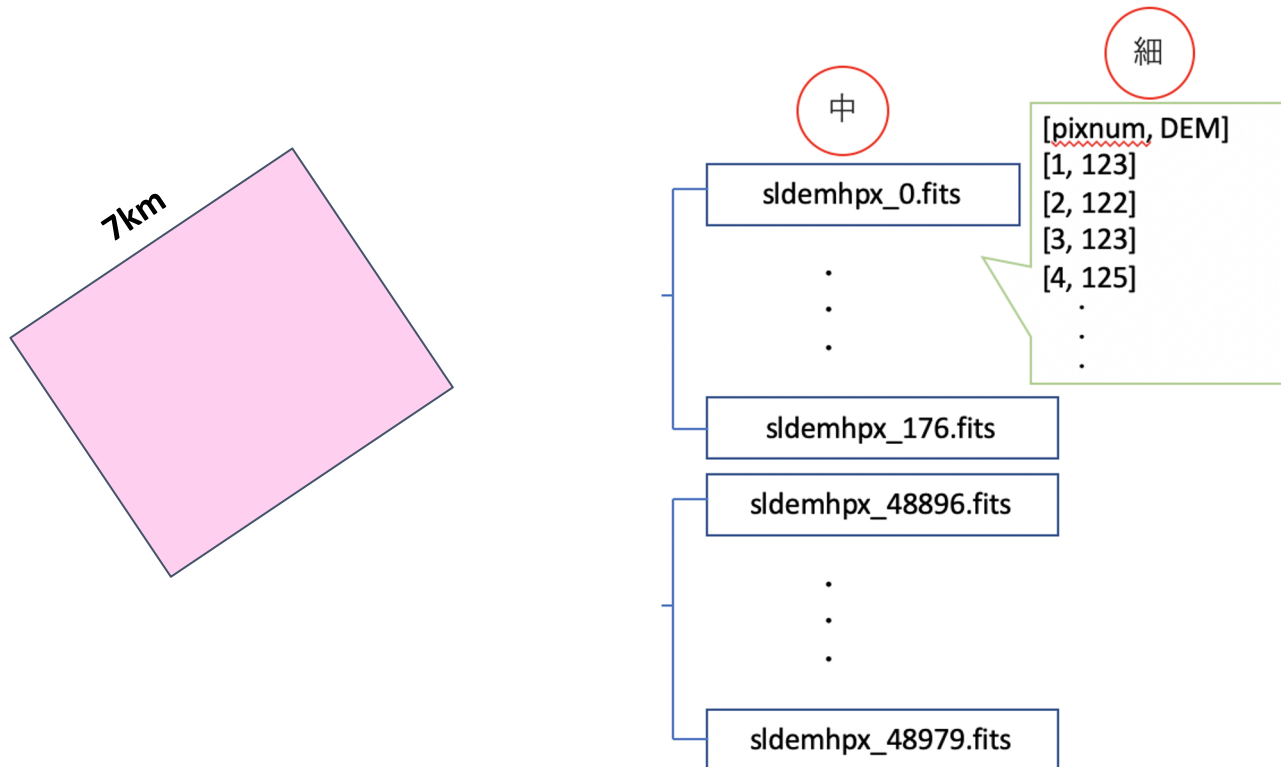


元々のDEMデータとほとんど同じ解像度になるようにNside 2^{18} に設定
→赤道付近でオーバーサンプリングになってしまう

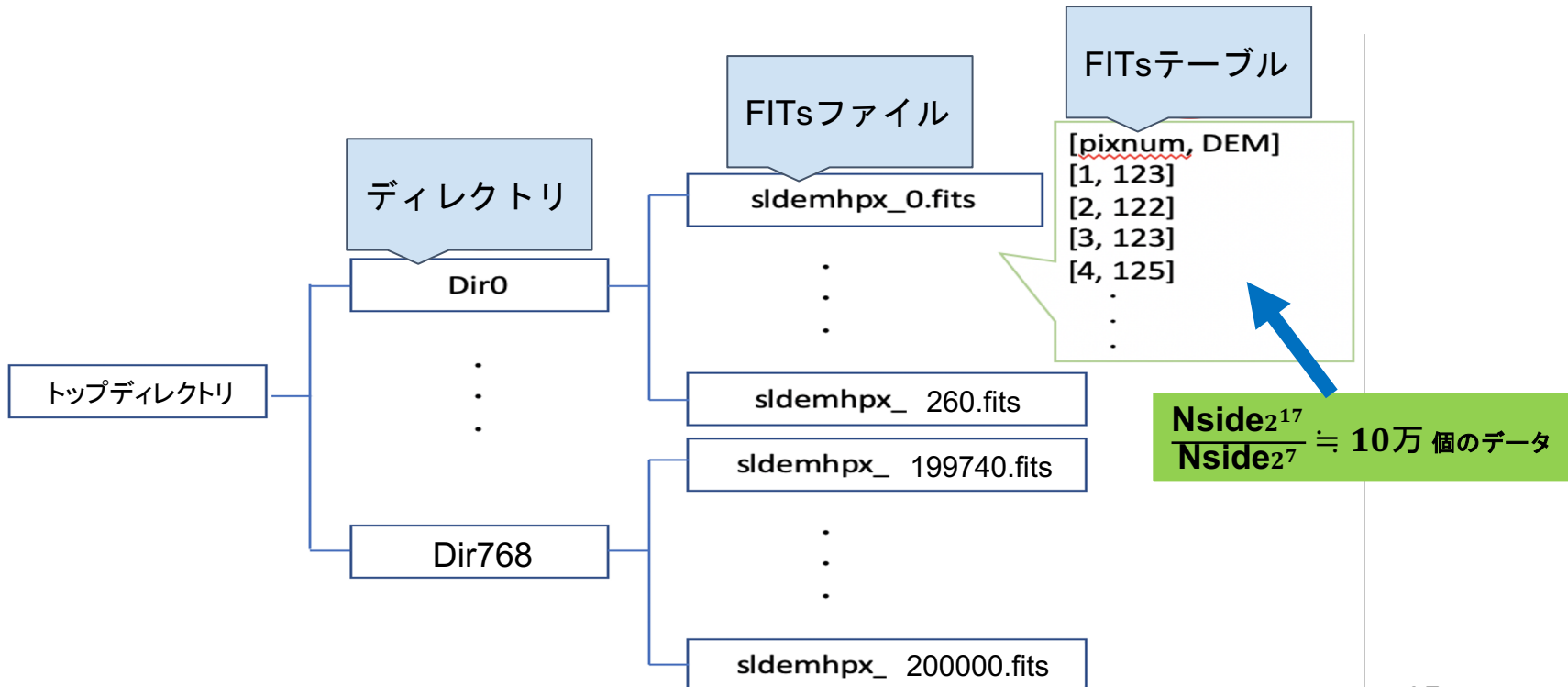
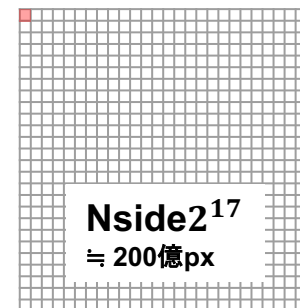
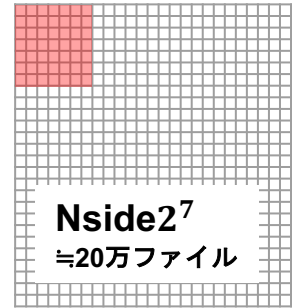
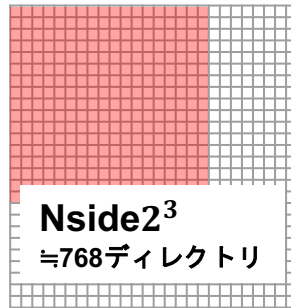
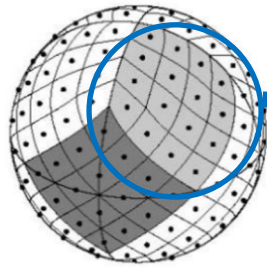
解像度を一つ下げた $N_{\text{side}}(\text{分割数}) = 2^{17}$ でHEALPix化を行なった
(全球を200億個のピクセルに分けた)

FITsファイルの分割数決定

- 使いやすいファイルの分割数の決定
 - 全天を $N_{\text{side}} = 2^7$ (約78万個)のファイルに分けた
 - 長辺の長さ : 約7km (地球の水平線は5km)



実際の階層構造

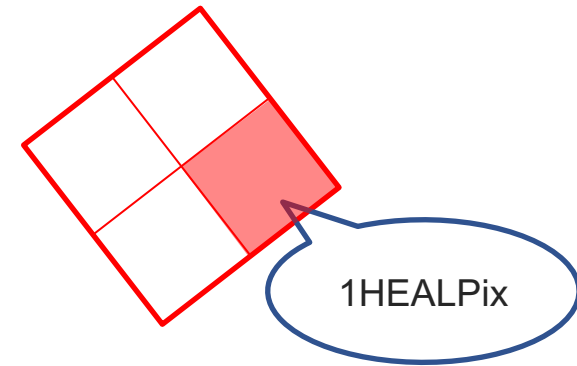


手法

1. DEMデータのHealPix構造化
2. データ作成の高速化
3. データ提供システムの初期検討

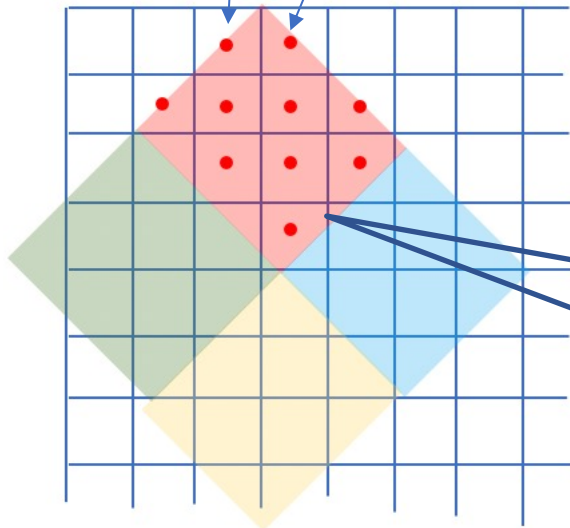
HEALPix化の具体的な流れ

- 月面高度データのHEALPix化
 - 1HEALPixごとに高度の値を平均化して算出
- 具体的な計算手法



{ピクセル番号, 高度の値} = {1, 10}

{2, -20}

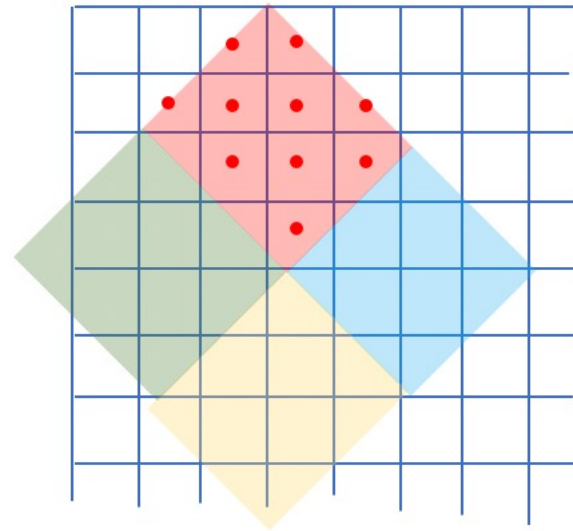
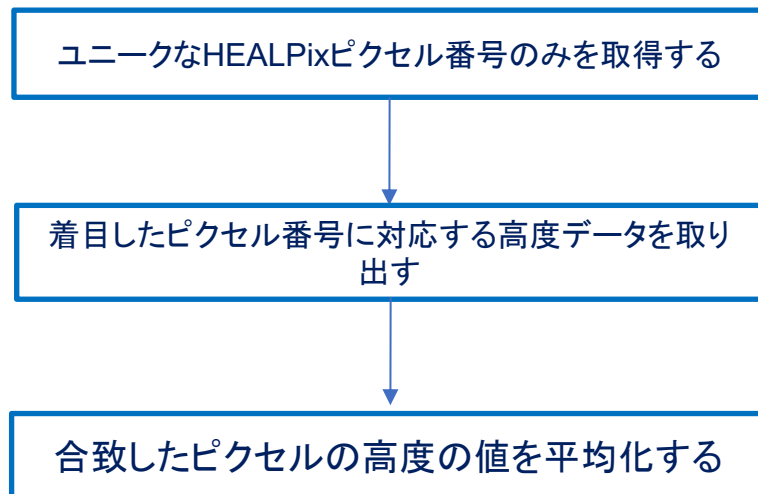


ピクセル番号	高度の値
1	10
2	-20
2	-24
3	50
3	52
7	48
10	100

1HEALPix内の高度の平均を取る
→ $\{10+(-20)+(-24)+50+52+48+100\} / 10$
=21.6

高速化前の処理

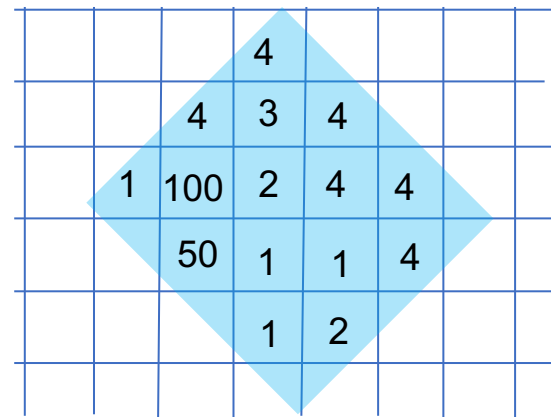
- for文を使用



解像度を概ね維持して変換するため、 4096×4096 の画像を $\sim 4096 \times 4096$ 回近い回数、走査する必要があり実行時間が長くなる

処理の高速化

- Tensorflowのsparse.bincount関数を用いて高速化する
 - Numpyのbincount()という値ごとの個数を要素とする配列を返す関数を利用できるが、全球のHEALPixピクセル数に対応する長さの配列(~2TB)の確保が必要となり実行できない
 - 全球の1/60000の領域しか着目せず、疎な行列となるため、現れたピクセル分のメモリのみを確保するsparse.bincount関数を使用することでメモリ破綻を防ぐことが可能



実際にはピクセル番号と高度から成る2次元配列であるが、簡易的に高度の値のみに着目すると

元の配列は

```
[4, 4, 3, 4, 1, 100, 2, 4, 4, 50, 1, 1, 4, 1, 2]
```

bincoutを使用すると

```
[4, 2, 1, 5, ..., 1, ..., 1]
```

sparse.bincountを使用すると

```
[4, 2, 1, 5, 1, 1]
```

高速化の比較

- $N_{\text{side}}2^{14}$ の時の時間で比較した

- for文を使用した時

- 1回目 → 1818.21秒
- 2回目 → 1771.65秒
- 3回目 → 1829.64秒

→ 平均1806.5秒

- sparse.bincountを使用した時

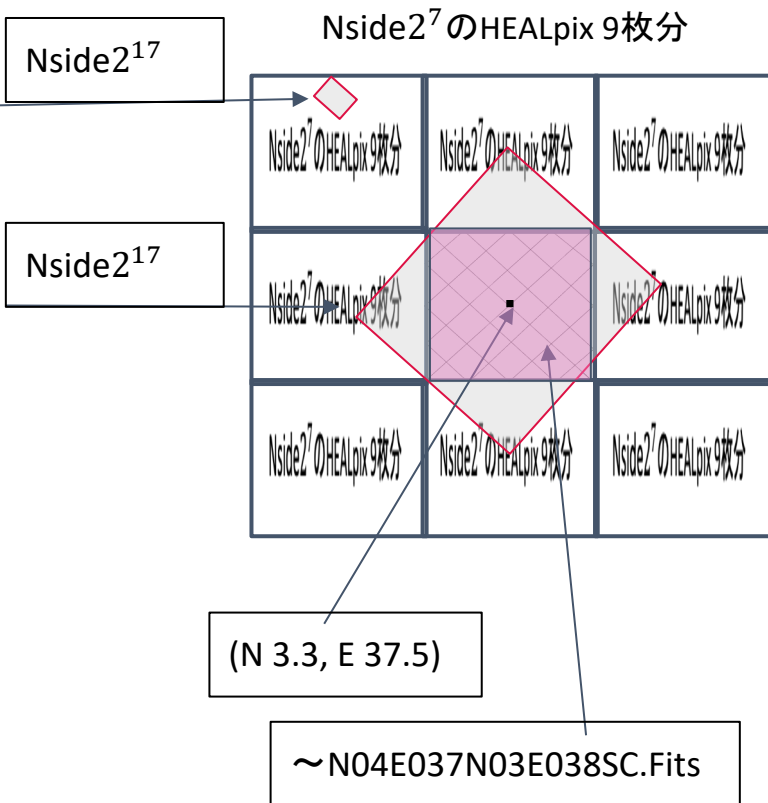
- 1回目 → 0.32秒
- 2回目 → 0.32秒
- 3回目 → 0.30秒

→ 平均0.31秒

より効率的なデータ処理のために

- 高速化

- 全球のDEMデータを一度に読み込むとデータ量が膨大



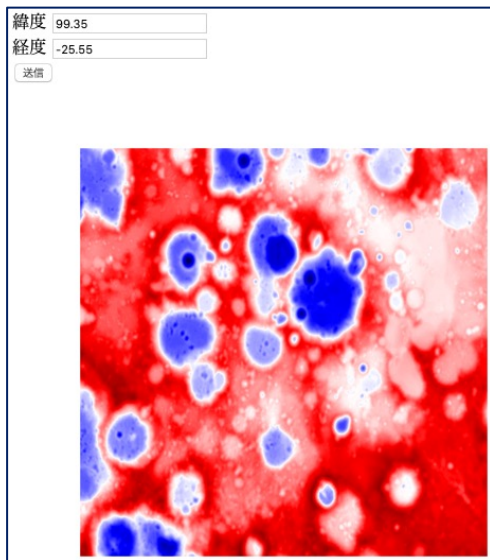
- ① Nside2⁷のhealpixの中心の緯度経度を取得
- ② 中心の緯度経度からFITSファイル名を検索
(ファイル名には緯度経度が含まれている)
~N04E037N03E038SC.Fits
- ③ 周囲8枚のFITSファイル名を取得
- ④ FITSファイルを読み込む
- ⑤ 分類したいNside2⁷のhealpix番号に対応するNside2¹⁷のピクセルを取得してFITSにまとめる

データ提供システムの初期検討

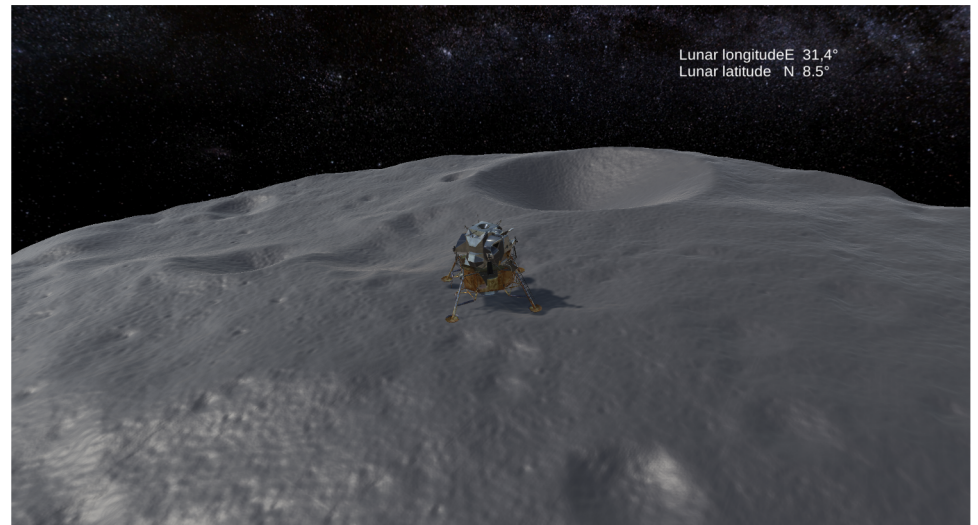
1. DEMデータのHEALPix化
2. データ構造について
3. データ作成の高速化
4. データ提供システムの初期検討

データ提供システムの初期検討

- Webアプリケーション
- ゲーム/3DCGへのデータ活用



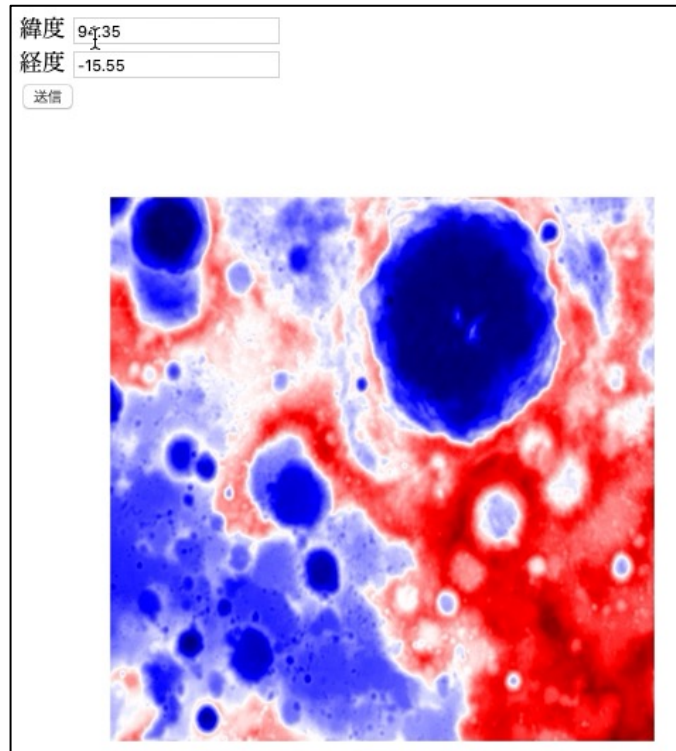
Webアプリケーション



ゲーム/3DCGへのデータ活用

Webアプリとしての応用

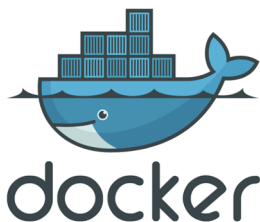
- 入力した緯度・経度の周辺のピクセルの可視化するWebアプリを開発し，動作を確認した



デモ動画

システム図

- Python用のWebアプリフレームワークであるFlaskを使用して実装した



フロントエンド

- 緯度・経度の入力
- DEMデータの出力



バックエンド

- 緯度・経度からDEMデータの取得



Flask

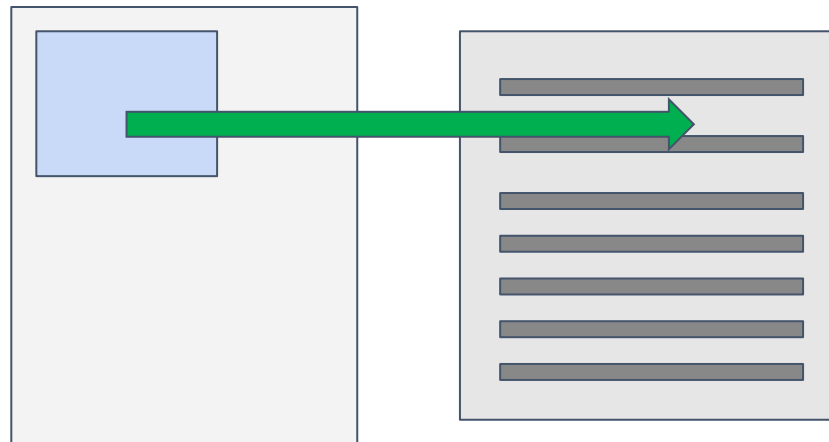
データベース

- HEALPix化したDEMデータ



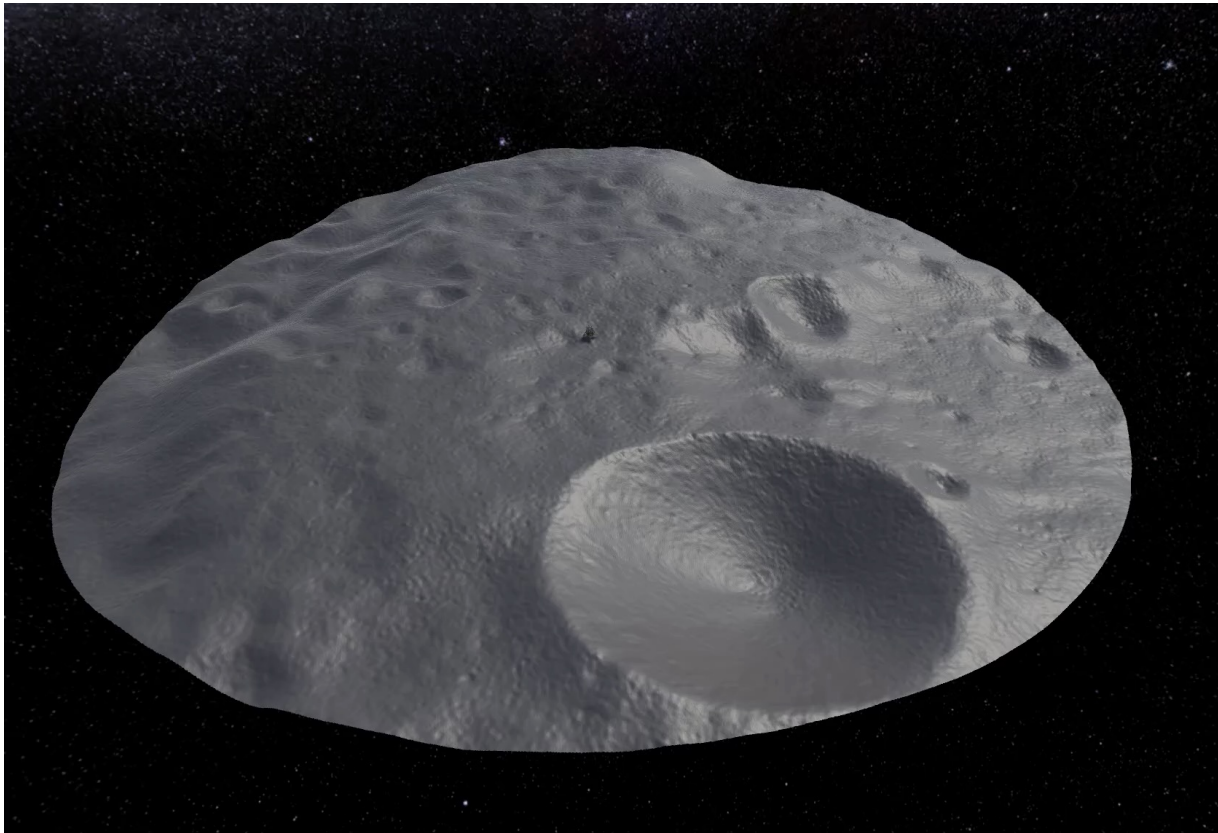
今後やりたいこと

- ユーザーが選択した範囲のDEMデータのリストを取得できるような機能の実装
 - 現在は数字で入力しているためGUIで取得できるように変更する



ゲーム / 3DCGへの応用

- 今回作ったデータをもとにメッシュを作成した
 - apollo11号の着陸地点

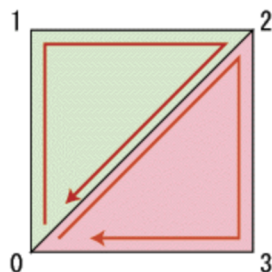


Apollo Lunar Model : NASA

点群データからのメッシュ化(1/2)

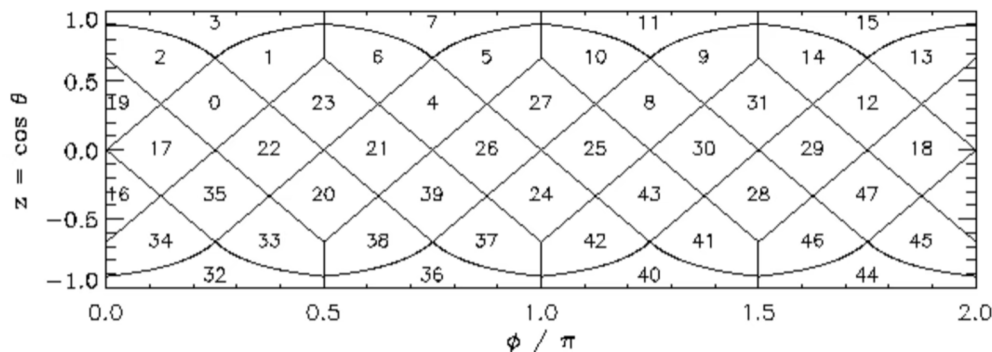
- メッシュの生成

- メッシュ化には点群データの点の並び方が非常に重要



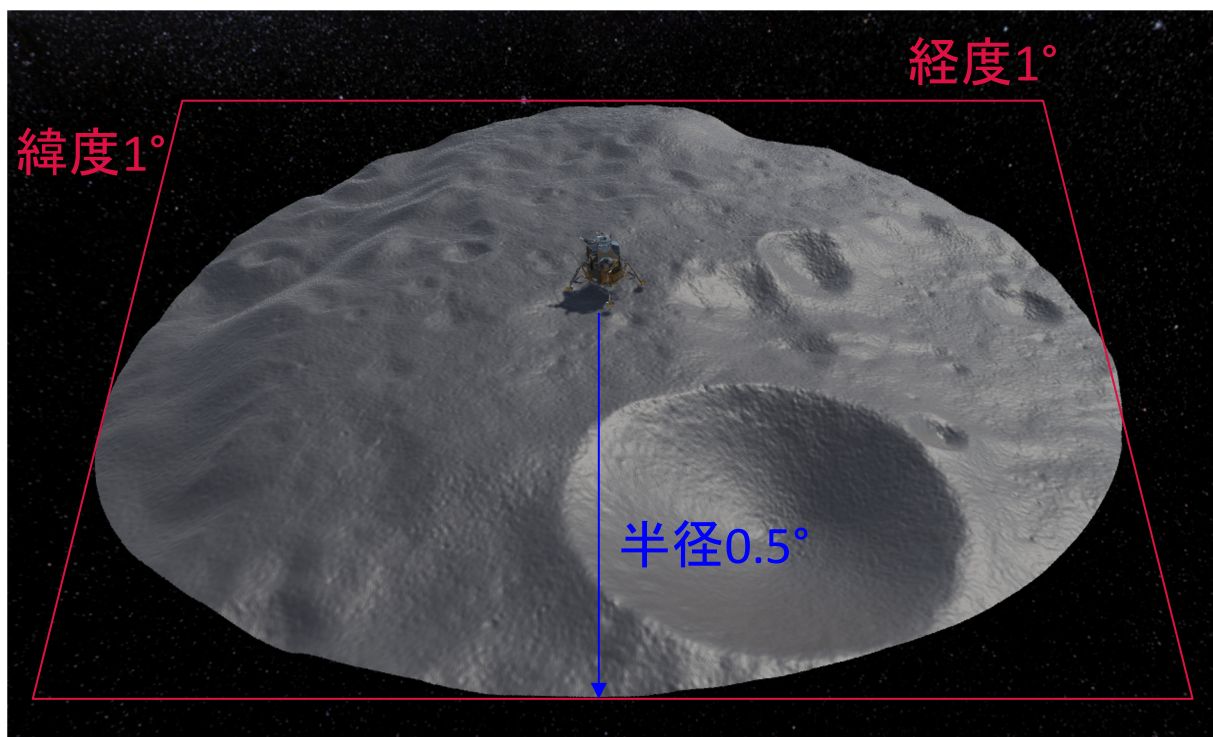
- 現在のHEALpixの並び方は NEST

- 点をつなぐ順番を決めるのが難しい



HEALpix化のメリット

- ダウンロードできる最小単位が小さくなったことで、
欲しい場所のDEMデータをピンポイントで取得できる



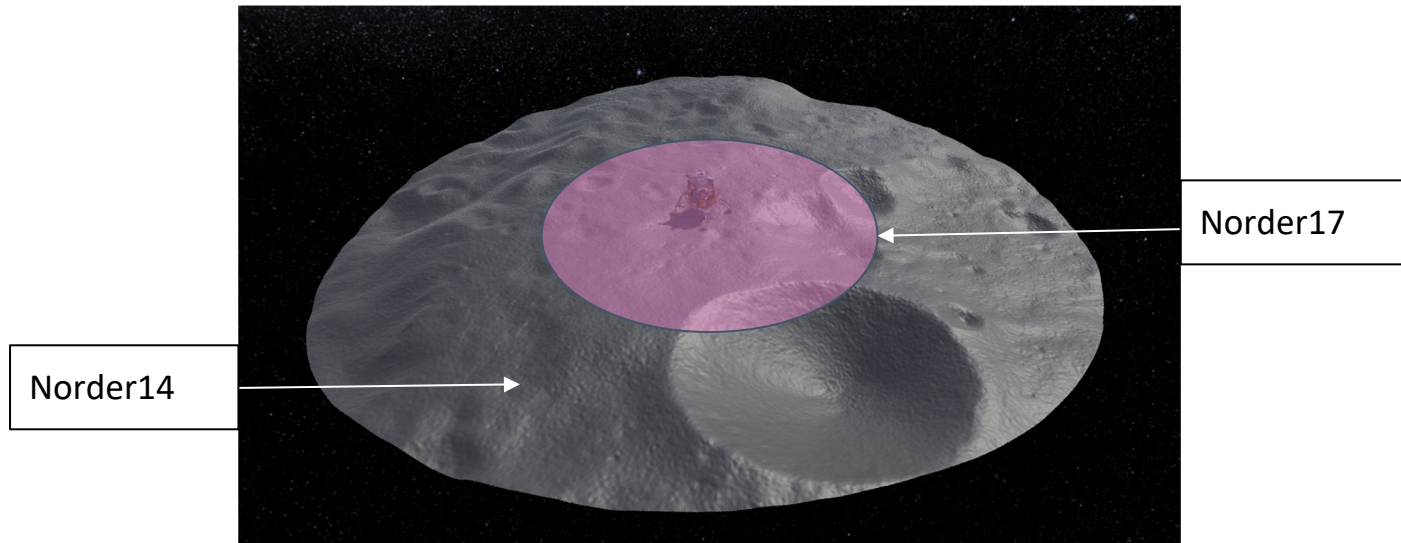
解像度 : $N_{side}2^{17}$
半径 : 0.5°
処理時間 : 10秒程度
データサイズ : 350MB



WEBサイトから
ダウンロード可能なサイズ

より使いやすい形にするには

- 用途に応じて解像度の違うデータをDLできる
 - 良い解像度のデータを使うと重い



- WEBサービスに組み込んでDartsから公開

まとめ

- 目的
 - かぐやの大規模データを簡単に利用できるようにデータフォーマットを変換すること
- 目標
 - HEALPixを用いた月面標高データベースを構築
- 結果
 - HEALPixベースのDEMデータを構築でき、任意の座標の高度データを即座に取得可能
 - 着目した緯度・経度の周辺の画像を即座に取り出すWebアプリを開発した
 - HEALPixの点群データからメッシュ化するアルゴリズムを開発した
- 今後の展望
 - Dartsから月面標高データへ簡易にアクセス・ダウンロード可能なWebアプリを公開する