



ISSN 2433-2216(Online)
JAXA-RR-18-008

宇宙航空研究開発機構研究開発報告

JAXA Research and Development Report

宇宙科学情報解析論文誌 第八号
Journal of Space Science Informatics Japan Volume 8

2019年3月

宇宙航空研究開発機構

Japan Aerospace Exploration Agency

目次

かぐや DEM を用いた, 機械学習による中央丘クレーター識別	1
1 はじめに	2
2 提案手法	3
2.1 クレーターの抽出	3
2.2 機械学習による識別	6
2.3 適合率と再現率	7
3 実験結果	7
3.1 データセット	7
3.2 パラメータ	8
3.3 学習モデル	8
3.4 識別精度	9
4 まとめ	10
参考文献	10
機械学習を用いた宇宙機の故障の予兆検知	11
1 はじめに	11
2 対象データ	12
3 異常検知手法	13
3.1 kNN を用いた異常検知	14
3.2 RNN を用いた異常検知	14
3.3 AutoEncoder を用いた異常検知	15
3.4 OneClassSVM を用いた異常検知	16
4 異常検知結果・考察	16
4.1 異常検知結果	16
4.2 考察	18
5 まとめ	19
参考文献	20
背面投影型デジタル地球儀 Dagik Earth のための半球面マルチタッチパネルの開発	21
1 背景および目的	22
2 タッチパネルの仕組みと原理	22
2.1 光波におけるスネルの法則	22
2.2 赤外 LED とブラケット	23
2.3 赤外線 Web カメラ	23
2.4 タッチ箇所の検出とポインティングデバイス制御の紐づけ	24
3 実験および考察	25
4 まとめ	25
謝辞	25
参考文献	25

「はやぶさ2」ハードウェアシミュレータに係る レイトレーシングソフトウェアの開発と評価	27
1 まえがき	27
2 OIS	28
2.1 OIS が満たすべき要件等	28
2.2 OIS ソフトウェア	29
2.3 画像生成の選択肢	30
3 レイトレーシング	31
3.1 ソフトウェアレイトレーシングの採用	31
3.2 Voxel 分割	31
4 評価	33
4.1 評価環境	33
4.2 取り扱うデータとレンダリング条件	33
4.3 Voxel 分割の評価	34
4.4 レンダリングの所要時間に係る評価	36
5 あとがき	40
参考文献	40

かぐや DEM を用いた, 機械学習による中央丘クレーター識別

原 聡志^{*1}, 山本 幸生^{*2}, 荒木 徹也^{*1}, 廣田 雅春^{*3}, 石川 博^{*1}

Identification of moon central peak craters by machine learning using Kaguya DEM

Satoshi Hara^{*1}, Yukio Yamamoto^{*2}, Tetsuya Araki^{*1}, Masaharu Hirota^{*3}
Hiroshi Ishikawa^{*1}

Abstract

There are many craters in the moon. Among them are craters having a special structure called “central peak” (hereinafter referred to as “the central-peak crater”). This central peak has an important characteristic that substances inside the moon crust are exposed on the moon surface. Therefore, by measuring the surface of the central peak, it is possible to estimate the material of the surrounding inner crust. By analyzing the inner crust, it is expected that estimation of the cause of craters and central peaks, the process of the environment of the moon surface, and crustal deformation of the past. However, except for some famous craters, the investigation has not progressed much. The reason for this is that the discovery of the central peak is based on visual observation of images by experts, so there are few known the central-peak craters. In order to solve this problem, it is necessary to automate the discovery method of the central-peak crater and prepare a catalog that records the position and size of central peaks, thereby greatly increasing the prospecting point candidate of the central-peak crater. Therefore, in this research, the final goal is to create a catalog of the central-peak crater, and for that purpose we propose an automatic discovery method of the central-peak crater. In this research, we use Digital Elevation Model (DEM) of the lunar surface observed by JAXA’s lunar orbit satellite “KAGUYA” to identify the central-peak crater by machine learning and verify its accuracy. Specifically, we first extract craters using a high-speed crater automatic extraction method called “Rotating Pixel Swapping Method for DEM”, label them, and then try to identify the central-peak crater by CNN. As a result, it was impossible to obtain a highly accurate discrimination model that could create the catalog of the central-peak craters, but we could confirm the possibility that CNN is an effective method in the central-peak crater identification.

Key Words : Moon Central Peak, Machine Learning, Neural Network

doi: 10.20637/JAXA-RR-18-008/0001

* 平成 30 年 12 月 19 日受付 (Received December 19, 2018)

^{*1} 首都大学東京大学院 システムデザイン研究科 (Tokyo Metropolitan University)

^{*2} 宇宙航空研究開発機構宇宙科学研究所 (Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency)

^{*3} 岡山理科大学 総合情報学部 (Okayama University of Science)

概要

月には大小多数のクレーターが存在している。その中には、「中央丘」という特殊な構造物を有するクレーター（以下「中央丘クレーター」と呼称）が存在している。この中央丘には、月面表面に表出しているながらも、月地殻内部の物質が露出しているという重要な特徴がある。すなわち、中央丘表面の探査によって、周囲の内部地殻の物質を推定することが可能となるのである。この内部地殻の分析により、クレーター及び中央丘の成因の推定や、過去の月面の表層環境や地殻変動の過程が推定できることが期待される。しかし現在、中央丘クレーターの探査が盛んに行われてはいない。その要因として、中央丘の存在確認が専門家の画像目視によるものであるが故、中央丘クレーターとして知られているクレーターが数少ないことが挙げられる。これを解決するため、中央丘クレーターの発見手法を自動化することで、中央丘クレーターの場所、大きさ等を網羅した一覧（本研究では中央丘クレーターカタログと呼称）を作成し、今後の月面探査において調査対象の候補となりうる中央丘を大幅に増加させることが求められる。よって本研究では、中央丘クレーターカタログの作成を最終目標とし、その為の中央丘クレーターの自動発見手法を提案する。本研究においては、JAXA の月周回衛星「かぐや (SELENE)」の観測にもたらされた月面の数値標高モデル (DEM) を用い、機械学習による中央丘クレーターの識別を行い、それが中央丘クレーターカタログを作成できるほどの精度を有しているか検証する。具体的には、まず回転ピクセルスワッピング法という DEM データからの高速クレーター抽出手法を用いて各クレーターの DEM データを抽出、それらにラベル付けを行った後、CNN による中央丘クレーターの識別を試す。結果として、中央丘クレーターカタログを作成できるほどの高精度な識別モデルを得ることはできなかったが、中央丘クレーター識別において CNN が有効な手法であることは確認できた。

1 はじめに

月面のクレーターの中には、内部に中央丘と呼ばれる構造物を有するものがある。この中央丘とは、クレーターの内部に形成される丘状の凸構造物で、月地層の内部物質が表層に露出している月面における貴重な観測点である。中央丘は主に生成時の衝突の規模が大きいもの、つまりは大型のクレーターに存在する傾向にあり、中央丘を有するクレーターは中央丘クレーターと呼称されている [1][2]。そして Matsunaga らの研究 [3] のように、この中央丘クレーターの分析によって、月の科学的解析における有力な情報を得ることが期待できる。しかし現在、中央丘の分析を推進する上で、中央丘クレーターのカタログが存在しないという課題がある。従来、分析対象とされてきた中央丘クレーターは、月面画像等を専門家が目視することで発見してきた。しかし、無数に存在する月面クレーターを一つ一つ専門家が確認し、それによって中央丘クレーターを網羅的に探索する事は長大な時間を要し、非効率的である。そこで本研究では、中央丘クレーターを自動で抽出する為の手法の考案を試みる。

専門家が月面クレーターから中央丘クレーターを識別する上で識別対象とするものはクレーターの画像であり、その画像情報は中央丘クレーターの形状を表している。この画像から中央丘クレーターに特有の構造を確認することで、専門家は目視での中央丘クレーター識別を行うことができる。その為、中央丘クレーターの構造的特徴を何らかの方法で解析することにより、中央丘クレーターの識別が自動化できると考えられる。そこで本研究では、JAXA の月周回衛星「かぐや (SELENE)」によって収集された月面の数値標高モデル (Digital Elevation Model, DEM)[4] を用いる。DEM は地形、本研究ではクレーターの形状を表現できる為、その解析によって中央丘クレーターの構造的特徴の解析が可能となる。本研究で用いる DEM データは全て、DARTS at ISAS/JAXA にて公開されている*4ものを使用する。

中央丘クレーターの構造的特徴の自動的な解析を実現するために、本研究では、月面物性や地質学への専門性を必要としない、機械学習による分類問題への帰着を提案する。機械学習による分類の為、我々はまず既存手法によるクレーターのみ抽出を行なう。そして学習のためのラベルをそれらに付与し、機械学習による中央丘クレーターの識別を行い、その有効性を検討する。

本論の構成は次の通りとなる。2章では、本研究及びその中で用いられる手法に関連する研究について言及

*4 DARTS at ISAS/JAXA
<http://darts.jaxa.jp>

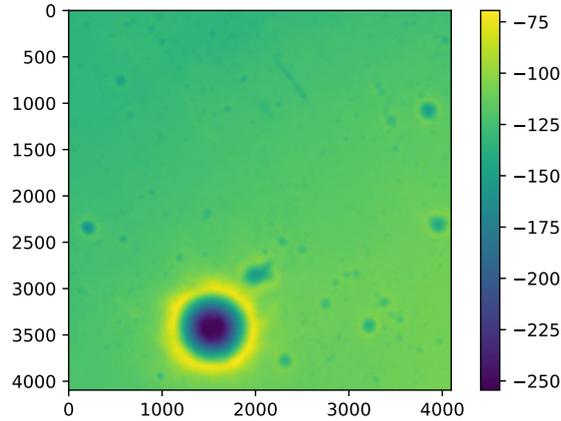


図 1: DEM データ

する。3章では、本研究における具体的な手法を提案する。4章では、3章での提案手法に基づいた実験を行い、その結果の分析、考察を行う。5章では、本論のまとめ及び今後の課題について述べる。

2 提案手法

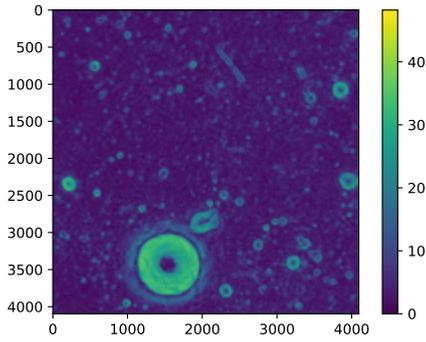
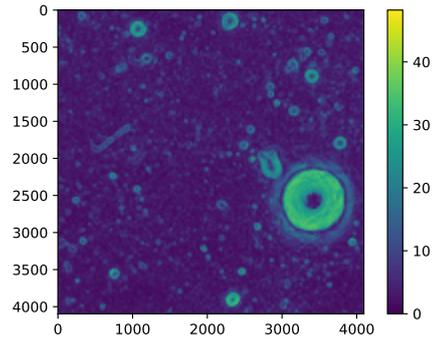
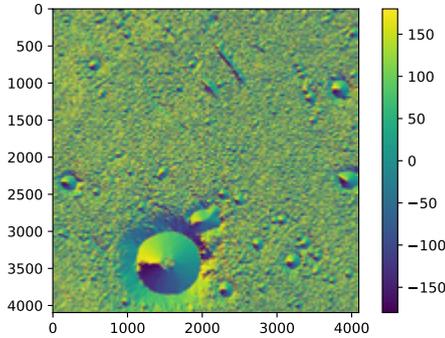
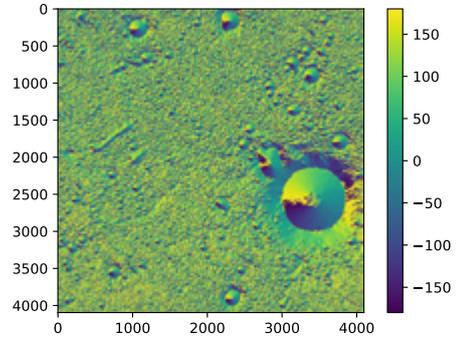
2.1 クレーターの抽出

本研究での前処理として、月面の DEM データからクレーターの領域のみを抽出する。図 1 に、DEM データの一例を示す。縦軸は南北 (単位はピクセル)、横軸は東西 (同左)、色は標高 (単位は m (メートル) をピクセルの値に変換) を表している。そこで本研究においては、山本らの開発した RPSD(RPSW for DTM) 法 [5] を用いてクレーターの自動抽出を行う。

RPSD 法の基本方針として、任意のある領域の DEM を解析対象とし、その各地 (ピクセル) の高度及び勾配、そしてクレーター内部の勾配の回転対称性に着目する。これらより解析対象の中からクレーターの中心となる地点を推定し、そこから周囲の標高を分析することで、具体的なクレーター範囲を決定していく。

具体的な処理として、まず解析対象となる一定範囲の DEM データを取得する。次に、取得した DEM データから、各地点の勾配の大きさを表す斜面勾配マップ $S(x, y)$ と各地点の勾配の向きを表す斜面方位マップ $F(x, y)$ を計算する。次に、 $S(x, y)$ と $F(x, y)$ について、それらを角度 ϕ だけ回転させたマップ $S_\phi(x, y)$ と $F_\phi(x, y)$ を作成する。本研究においては、 $\phi = 90^\circ, 180^\circ, 270^\circ$ のものを作成した。図 2~図 5 に、斜面勾配マップ及び斜面包圍マップの一例を示す。解析範囲は図 1 と同じ範囲であり、斜面勾配マップの色は勾配の大きさを、斜面包圍マップの色は勾配の向き (0° で東、 180° もしくは -180° で西、 -90° で南、 90° で北向きに上がっていく) を表す。

以上を作成した後、ある一点のピクセル (x_0, y_0) における回転対称性を計算していく。まずは、中心点 (x_0, y_0) から縦軸、横軸とも $\pm l_{max}$ (抽出対象最大サイズ指定のためのパラメータ) の範囲を $S(x, y), F(x, y)$ から切り出し、 $S_{x_0, y_0}(x, y)$ 及び $F_{x_0, y_0}(x, y)$ を作成する。同様の範囲での切り出しを $S_\phi(x, y), F_\phi(x, y)$ でも行い、 $S_{x_0, y_0, \phi}(x, y)$ 及び $F_{x_0, y_0, \phi}(x, y)$ を作成する。次に、作成した $S_{x_0, y_0}(x, y)$ 及び $S_{x_0, y_0, \phi}(x, y)$ より、斜面勾配が $\theta_L \sim \theta_U$ の範囲にあるピクセルだけを抽出した斜面勾配抽出マップ $U_{x_0, y_0, \phi}(x, y)$ を作成する。また、 $F_{x_0, y_0}(x, y)$ 及び $F_{x_0, y_0, \phi}(x, y)$ より、斜面方位について回転角 ϕ に対して回転対称となっているピクセル (閾値 ω

図 2: 斜面勾配マップ $S(x, y)$ 図 3: 90° 回転後の斜面勾配マップ $S_{90}(x, y)$ 図 4: 斜面方位マップ $F(x, y)$ 図 5: 90° 回転後の斜面方位マップ $F_{90}(x, y)$

を設定し、回転時の斜面方位の差が ω 以下となるようなピクセル)だけを抽出した回転対称点抽出マップ $V_{x_0, y_0, \phi}(x, y)$ を作成する。 $U_{x_0, y_0, \phi}(x, y), V_{x_0, y_0, \phi}(x, y)$ の両マップにおいては、抽出されたピクセルの値を 1, 抽出されなかったピクセルの値を 0 とおく。

そして、全ての $U_{\phi}(x, y)$ 及び $V_{\phi}(x, y)$ において値が 1 となっているようなピクセルのみを抽出した回転対称性評価マップ $H_{x_0, y_0}(x, y)$ を作成する。このマップはすなわち、斜面勾配が $\theta_L \sim \theta_U$ の範囲にあり、かつ斜面方位が $360/\phi$ 個の回転角に対して回転対称条件を満たすピクセルだけを抽出したマップである。図 6～図 11 に、回転対称性を持つ点と持たない点それぞれの斜面勾配抽出マップ、回転対称点抽出マップ、回転対称性評価マップを示す。黄色いピクセルが抽出された点、紫のピクセルが抽出されなかった点を示す。

この $H_{x_0, y_0}(x, y)$ の全ピクセルの値を合計し、同様の計算を解析対象 DEM データの全ピクセルで行う (実際には計算コスト低減のため、数ピクセルおきに計算を行う) ことにより、解析対象 DEM データに対する回転対称性関数 $R(x, y)$ を求める。そして、 $R(x, y)$ の高いものから閾値分率 f を使ってクレーター中心候補点をリストアップする。

次に、リストアップされた候補点に対し、クレーター径の測定を行う。まず、中心候補点から横軸、縦軸それぞれの正負方向への高度プロファイル $P(n)$ を作成する。そして $P(n)$ よりさらに斜面勾配プロファイル $Q(n)$ を作成する。これらより、 $n = l_{min}$ (抽出対象最小サイズ指定のためのパラメータ) から n を順次増加させていき、中心候補点からの高度が P_{min} (最小深さ閾値) より高く、かつクレーター壁面の最大勾配から σ だけ勾配が緩くなった (もしくは勾配の値が負になった) 最初の地点での n を求める。この際、 n が l_{max} に達

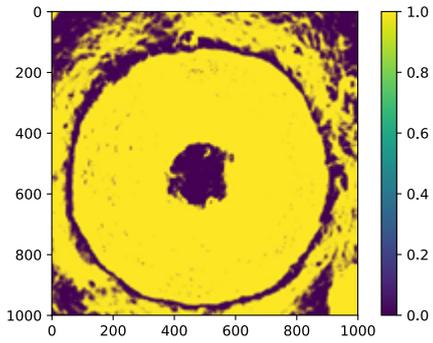


図 6: 回転中心性のある観測点における
斜面勾配抽出マップ $U_{x_0, y_0, \phi}(x, y)$ ($\phi = 270^\circ$)

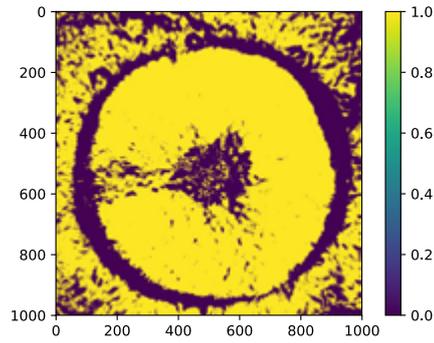


図 7: 回転中心性のある観測点における
回転対称点抽出マップ $V_{x_0, y_0, \phi}(x, y)$ ($\phi = 270^\circ$)

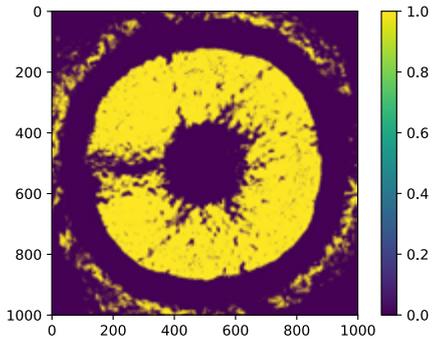


図 8: 回転中心性のある観測点における
回転対称性評価マップ $H_{x_0, y_0}(x, y)$

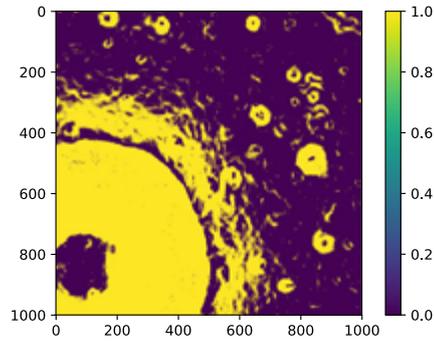


図 9: 回転中心性のない観測点における
斜面勾配抽出マップ $U_{x_0, y_0, \phi}(x, y)$ ($\phi = 270^\circ$)

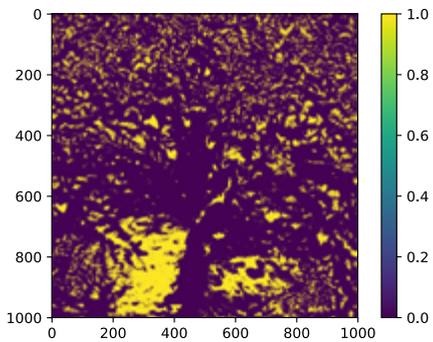


図 10: 回転中心性のない観測点における
回転対称点抽出マップ $V_{x_0, y_0, \phi}(x, y)$ ($\phi = 270^\circ$)

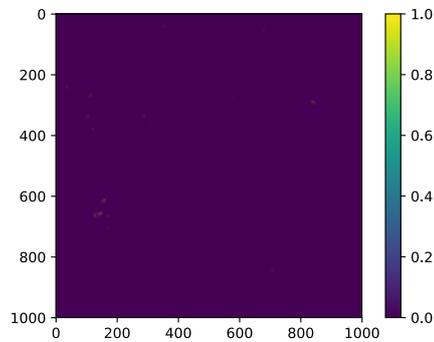
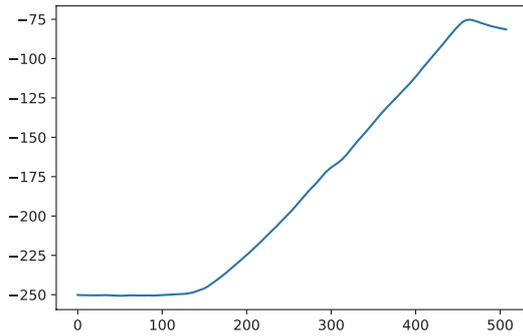
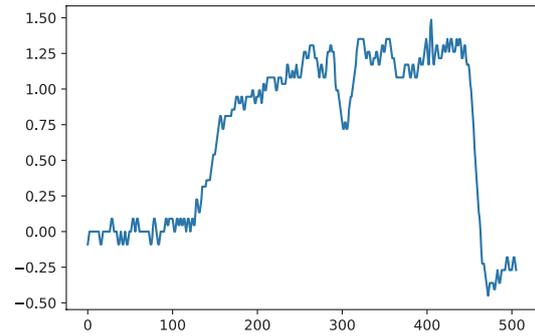


図 11: 回転中心性のない観測点における
回転対称性評価マップ $H_{x_0, y_0}(x, y)$

図 12: 横軸正方向への高度プロファイル $P(n)$ 図 13: 横軸正方向への斜面勾配プロファイル $Q(n)$

しても条件を満たさなかった場合、計算結果は導出しない。 P_{min} 及び σ については、山本らの研究に準じ、 $P_{min} = l_{max} * 0.05$ 、 $\sigma = 15$ とした。以上の計算作業を中心候補点の上下左右四方向に行い、導出された数値の平均を中心候補点に対するクレーターのリム径 (クレーター径) とした。この際、任意の整数パラメータ s_{min} ($0 < s_{min} < 5$) を設定し、計算結果が s_{min} 個以上導出されなかった候補点については、クレーター中心ではなかったとみなす。図 12 及び図 13 に、横軸正方向 (東向き) の高度プロファイル及び斜面勾配プロファイルを示す。対象点は図 6 と同じ点で、横軸がピクセル数、縦軸が標高 (単位は m をピクセルに変換) もしくは勾配 (単位は rad) を表す。

以上の計算により、一つの対象 DEM データ内に存在する半径 $l_{min} \sim l_{max}$ のクレーター群が抽出される。これらを DEM データそれぞれに実行することにより、クレーター群のデータセットを得る。

2.2 機械学習による識別

次に、抽出したクレーター群を機械学習によって分類し、中央丘クレーターを識別する。抽出されたクレーターの DEM は 1 チャンネルの、即ち各ピクセル毎に一つだけ値を有する画像データと類似したデータ形式である為、画像処理分野において用いられる機械学習手法が有効であると考えられる。よって、画像分類の代表的な手法の一つである CNN(Convolution Neural Network)[6] による分類を行う。CNN とは教師あり学習、つまり教師データを必要とする学習であり、機械学習アルゴリズムの一つである Neural Network に畳み込みフィルタの層を導入したもので、画像識別において高い性能を発揮することが示されている。

CNN の動作の概要は以下の通りである。CNN に入力された画像はまず、畳み込み層、プーリング層の二種を任意に積層させたネットワークを通る。畳み込み層とは、入力画像に対しフィルタによる畳み込みを行う層である。簡単のために入力、出力ともチャンネル数 1、即ち各ピクセル毎に一つしか画素値を持たない二次元画像で考えると、入力画像を $I_{(x,y)}$ 、フィルタのカーネル (フィルタを表す行列、ここでは大きさを 3×3 とする) を $K_{(x,y)}$ 、出力画像を $O_{(x,y)}$ とした時の畳み込みの計算は以下ようになる。

$$O_{(x,y)} = \sum_{j=1}^3 \sum_{k=1}^3 I_{(x-2+j,y-2+k)} \times K_{(j,k)}$$

この演算によって、畳み込み層では入力画像からエッジ等の特徴を抽出できる (y 軸方向のエッジを抽出するフィルタのカーネルの一例: $[[1, 2, 1], [0, 0, 0], [-1, -2, -1]]$)。カーネルにチャンネル数の次元を付加して三次元行列とすれば、チャンネル毎に別々の特徴が抽出され、出力は各ピクセル毎に複数の値を有する三次元画像となる。プーリング層とは、入力画像の圧縮を行う層である。畳み込み層、プーリング層によって特徴量の抽

出と次元削減を行った後は、Neural Network と同様の全結合層、即ち (多層) パーセプトロンによって分類を行う。以上のネットワークにおいて、畳み込み層のカーネルおよび全結合層の重み・バイアスを変数として教師データによって学習させることで、目的の識別モデルを得ることができる。

CNN による分類の為、識別モデル作成の前に人手によるクレーター群へのラベル付けを行う。ラベル付けの際は、抽出したクレーターの DEM を二次元画像として表示し、それを実験者が目視確認することによって、どのラベルを付与すべきかの判断を行った。付与するラベルは、「クレーターでない」「非中央丘クレーターである」「中央丘クレーターである」の三種とした。「クレーターでない」というラベルを設けた理由は、より網羅的な中央丘クレーター探索を行う為、RPSD 法において後述のようにパラメータを調整し、高再現率低適合率なデータセットを作成したためである。加えて、中央丘の有無に関係しない、CNN でのクレーターの識別可能性についても検証するためでもある。

2.3 適合率と再現率

本研究において精度評価の指標として用いる、適合率および再現率について説明する。任意のラベル A に関する適合率、再現率はそれぞれ以下の式で計算される。

$$\text{適合率} = \frac{\text{機械学習によって } A \text{ と識別され、正解のラベルが } A \text{ であるデータの数}}{\text{機械学習によって } A \text{ と識別されたデータの総数}}$$

$$\text{再現率} = \frac{\text{機械学習によって } A \text{ と識別され、正解のラベルが } A \text{ であるデータの数}}{\text{正解のラベルが } A \text{ であるデータの総数}}$$

適合率は、実験結果がどれ程誤りのない結果を示しているかを測る指標である。一方再現率は、実験結果が本来現れるべき結果をどれ程漏らさずに示しているかを測る指標である。一般にこの二指標はトレードオフの関係にある。つまり適合率を上げようとするほど再現率は低下し、逆に再現率を上げようとするほど適合率は低下する。本研究では、中央丘クレーターに対する識別精度の高い識別モデルを構築することで、それによる中央丘クレーターの自動抽出を行うことを目的としている。よって、中央丘クレーター以外の物を中央丘クレーターとして誤識別される事や、中央丘クレーターでありながらそうと識別されない事より少ない、つまり中央丘クレーターに関する適合率・再現率の高い識別モデルを構築することが本研究の主目標である。

3 実験結果

3.1 データセット

本実験においては、ISAS/JAXA により公開されている月面の DEM データを用いる。本データは月面球表面を、「正距円筒図法 (Simple Cylindrical Projection)」と呼ばれる同径の円筒上に射影する投影法によって長方形に変形させた上で、横軸に経度方向すなわち東西、縦軸に緯度方向すなわち南北をとったものである。全球面は $1^\circ \times 1^\circ$ の小正方形に分割されており、小正方形一枚につき $4,096 \times 4,096$ ピクセル、その各ピクセルごとに高度データが格納されたバイナリデータとなっている。小正方形は全部で $360 \times 360 = 129,600$ 枚存在するが、南北の極点に近づくほど DEM の経度方向への歪みが大きくなっていきクレーター抽出が困難になっていく為、本研究においては極付近は実験対象とせず、北緯 $60^\circ \sim$ 南緯 60° 、東経 $180^\circ \sim$ 西経 180° を実験対象とした。

RPSD 法によって抽出されたクレーターへのラベル付けを施した結果、非クレーター及び非中央丘クレーターに対し中央丘クレーターの数が非常に不足していた。CNN での学習において教師データ内であるラベルのデータ数が相対的に極端に少ない場合、学習後の識別モデルがそのラベルの識別に関して極端に低精度になる危険がある。よって学習データセット内でのラベル毎のデータ数を均一にするため、中央丘クレーターにの

み反転, 回転等の処理を施し, データセットのオーバーサンプリングを行った.

前処理を施したデータセットは学習用データとテストデータに分割し, 学習用データで識別モデルの学習を行い, テストデータで識別精度の検証を行った.

3.2 パラメータ

実験に際し, USGS(アメリカ地質調査所)/NASA より公開されている^{*5}クレーターカタログ [7] を元に, RPSD 法におけるパラメータの一部を調整した. 調整対象となるパラメータは, クレーター抽出精度に対し影響の大きい f および s_{min} とした. 両パラメータ共に, 値を大きくする程抽出対象クレーターの形状に対して厳密性を要求するようになるが, 本研究では抽出されたクレーターの識別を CNN によって抽出後に行える為, この厳密性を緩める方向での調整を行った. これらは, f を小さくすることによって回転中心性の低さによって間引かれるクレーター中心候補点の数を減らし, より回転中心性の低い中心点を有するクレーター, つまり真円に対して歪みの大きいクレーターを許容し, s_{min} の値を減らす事でクレーター中心から東西南北いずれかの位置に別のクレーターによる「えぐれ」が存在した場合を許容する事で, 抽出されるクレーターの絶対数を増やす事を意図したものである. 両パラメータの検証, 調整はグリッドサーチによって行い, 結果として本研究において用いる値は $f = 0.003$, $s_{min} = 3$ とした. このパラメータによる抽出では, 抽出対象サイズのクレーターに対し約 80% の再現率を記録した. また, クレーター中心候補点の導出精度及び計算コスト削減のため, l_{min} , l_{max} は山本らの論文内で用いられていた値と比較して小さな値とした. より大型のクレーターを探索する場合も, この値を増大させるのではなく, 解析対象 DEM データをつなぎ合わせ, それを $4,096 \times 4,096$ に縮小することで抽出可能範囲に落とし込んだ. 以上のようにして, 本実験においては一回の RPSD 法計算における解析対象範囲を 32° 四方, 64° 四方, 128° 四方の三通り, (l_{min}, l_{max}) をそれぞれ $(8, 32)$, $(8, 32)$, $(16, 64)$ とすることで, 抽出対象クレーターの直径を合わせて約 4km~128km とした. これ以上の大きさのクレーターに関しては, USGS のクレーターリストに記載の数値を利用し, RPSD 法を用いずに自動抽出した. 抽出されたクレーターの中で, 小型のクレーターは数が膨大であるのに対し中央丘クレーターの数が少ないため, 本実験では直径が約 8km 以上のクレーターのみを学習に利用した.

3.3 学習モデル

本実験で構築する CNN 学習モデルは以下の通りである.

入力に抽出された 500×500 のクレーターの DEM データ

⇒ フィルタ数 16 の畳み込み層及びプーリング層

⇒ フィルタ数 32 の畳み込み層及びプーリング層

⇒ フィルタ数 64 の畳み込み層及びプーリング層

⇒ フィルタ数 128 の畳み込み層及びプーリング層

⇒ ノード数 4,096, 活性化関数 Leaky Relu の全結合層 $\times 2$

⇒ ソフトマックス関数による三値分類

学習における最適化アルゴリズムには Adam[8] を用いた. 実装には深層学習用フレームワーク Tensorflow[9] を, 実験環境には NVIDIA GeForce 1080Ti を使用し, GPU 環境による学習を行った. 一世代あたり 50 枚の入力で学習を行い, 学習に一世代あたり約 1.32 秒を要した.

^{*5} USGS (アメリカ地質調査所) /NASA
<https://planetarynames.wr.usgs.gov/Page/MOON/target>

表 1: 識別結果

		正解ラベル				
		非クレーター	非中央丘	中央丘	再現率	適合率
テストデータ数		966	439	272		
識別結果	識別数 (非クレーター)	859	111	13	88.9%	87.4%
	識別数 (非中央丘)	32	133	32	30.3%	67.5%
	識別数 (中央丘)	75	195	227	83.5%	45.7%

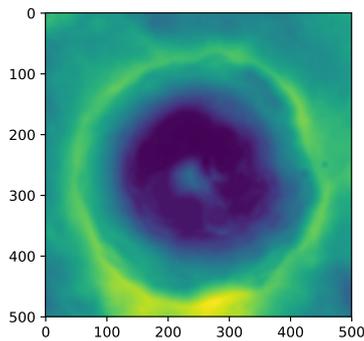


図 14: 中央丘クレーターの識別成功例

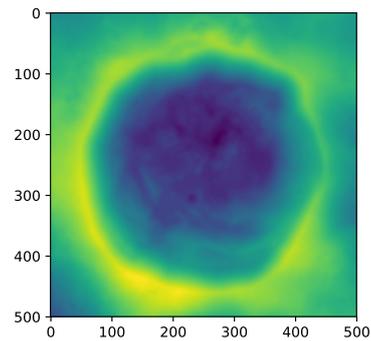


図 15: 中央丘クレーターの識別失敗例

3.4 識別精度

提案手法により, 3.3 節のモデルを用い, ミニバッチ数 50 の確率的勾配降下法 [10] によって 20,000 世代の学習を行った. 過学習防止のため, early stopping によって最適な学習モデルの保持を図った.

提案手法による中央丘クレーター識別結果は, テストデータにおいて表 1 のようになった. 表 1 において, 横軸のラベルはテストデータ群に付与した正解のラベル, 縦軸はテストデータ群を識別した時の識別結果である. これより, 中央丘クレーターの識別精度は再現率が 83.5%, 適合率が 45.7% であった. 特に適合率が低く, モデルとしての識別制度は中央丘クレーターカタログの作成に用いることができる制度に達していないと考える. 低適合率の要因について, 図 15 のようにクレーター内部に起伏があるようなクレーターを中央丘クレーターと誤識別する事例が多く見受けられた. これは, 教師データを人力で作成した際, 中央丘クレーター数の少なさを補うため, クレーター内に何らかの凸構造物が存在するようなクレーターを全て中央丘クレーターとしてラベル付けしていたことが一因と考えられる. 図 14 及び図 15 に, 分類モデルに中央丘クレーターとして分類されたクレーターの識別成功例及び誤識別例を示す. 一方, 中央丘の有無を考慮しないクレーターか否かの識別精度は, 再現率が 82.6%, 適合率が 83.3% であった. これより, クレーターか否かの識別に関しては CNN が示唆されている. 非クレーターでありながら中央丘クレーターとして誤識別される事例が特に多かったため, 出力をクレーターか否かの 2 ラベルのみとするような識別モデルを別に作成すれば, より高精度な識別モデルの実現が可能であると考えられる.

4 まとめ

本研究では、DEMデータの画像処理的特性に着目し、CNNによる中央丘クレーターの自動識別手法を検証した。実験の結果、現在の識別モデルによって未知の中央丘クレーターを高精度に網羅するにはまだ精度が不十分であったが、CNNが中央丘クレーターの特徴を抽出できている可能性は示唆された。

今後の課題として、RPSD法のグローバル対応による教師データの増加と月面に存在する全クレーターの網羅、及び教師データの再検討によるモデルの再学習、高精度化が挙げられる。また、提案手法をより拡張し、中央丘以外の月面構造物の分類、識別をDEM及びCNNを用いて行うことが、本研究のより発展的な課題である。

参考文献

- [1] Carlton C Allen. Central peaks in lunar craters. *Earth, Moon, and Planets*, Vol. 12, No. 4, pp. 463–474, 1975.
- [2] W Hale and JW Head. Central peaks in lunar craters-morphology and morphometry. *Lunar and planetary science conference Proceedings*, Vol. 10, pp. 2623–2633, 1979.
- [3] Tsuneo Matsunaga, Makiko Ohtake, Junichi Haruyama, Yoshiko Ogawa, Ryosuke Nakamura, Yasuhiro Yokota, Tomokatsu Morota, Chikatoshi Honda, Masaya Torii, Masanao Abe, et al. Discoveries on the lithology of lunar crater central peaks by selene spectral profiler. *Geophysical Research Letters*, Vol. 35, No. 23, 2008.
- [4] 坪内彩音, 篠田玲奈, 郭哲也, 後藤祐紀, 春山純一, 三宅互. Selene (かぐや) 搭載地形カメラステレオペアデータから得られた数値地形モデル (dtm) ならびに数値標高モデル (dem) プロダクトの標高値の検証報告. 宇宙航空研究開発機構研究開発資料, pp. 1–36, 2016.
- [5] 山本聡, 松永恒雄, 中村良介, 関根康人, 平田成, 山口靖. 回転ピクセルスワッピング法を使ったクレーター年代学. 日本惑星科学会誌遊星人, Vol. 24, No. 1, pp. 20–32, 2015.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.
- [7] LA Andersson and Ewen Adair Whitaker. Nasa catalogue of lunar nomenclature. 1982.
- [8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*, pp. 177–186, 2010.

機械学習を用いた宇宙機の故障の予兆検知

梅津 里香*¹, 杉江 卓哉*¹, 長瀬 雅之*¹, 湖海 亮*¹,
竹島 敏明*², 海老沢 研*², 満田 和久*², 山本 幸生*²

Detection of failure sign of spacecraft using machine learning

Rika Umezu*¹, Takuya Sugie*¹, Masayuki Nagase*¹, Ryo Kokai*¹,
Toshiaki Takeshima*², Ken Ebisawa*², Kazuhisa Mitsuda*² and Yukio Yamamoto*²

Abstract

Safety and reliability of the spacecraft are very important. It is desirable to predict possible failures of a spacecraft in advance and to have an operational environment leading to accident prevention. By applying failure analysis utilizing machine learning technology to the operation data of a spacecraft, we accumulate techniques to detect defects beforehand and are doing research to contribute to stable operation of the spacecraft. We tried to detect a sign of failure of the power supply using operational data of the X-ray astronomical satellite "Suzaku". As a result, we suggest that it is possible to detect a sign of failure under certain conditions.

Keywords: machine learning, failure analysis, spacecraft, Suzaku

概要

宇宙機は安全性や信頼性が厳しく問われ、未然に危険を予知し、事故防止に繋げる運用環境が望まれる。宇宙機の運用データに機械学習の技術を適用して故障解析を行い、不具合を未然に検知する技術を蓄積し、安定した宇宙機運用に資するための研究を行っている。

X線天文衛星「すざく」の運用データを用いた電源系機器の故障の予兆検知を試み、一定条件下での故障の予兆を検出することが可能であることを確認した。

1. はじめに

宇宙機は一度宇宙に打ち上げると修理を行うことが難しいため安全性、信頼性が厳しく問われる。そのため、日々の運用において未然に危険を予知し、事故防止に繋げる運用環境が望まれる。宇宙機は10年以上の長期運用が行われる場合も多く、宇宙環境という過酷な環境に晒される中で、運用中に故障する機器が少なくない。宇宙機の異常や故障の予兆を早期に検出することは、

doi: 10.20637/JAXA-RR-18-008/0002

* 平成30年12月19日受付 (Received December 19, 2018)

¹ 株式会社セック (Systems Engineering Consultants Co., LTD.)

² 宇宙航空研究開発機構宇宙科学研究所 (Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency)

安定運用のための重要な要件である。

長期運用される宇宙機には、膨大な運用データ・観測データが存在する。膨大なデータから得られる運用中に発生した出来事と運用データの関係性を学習することで、機器の故障を事前に捉えられる可能性があると考えられる。

本研究では、X 線天文衛星「すざく」の運用データを使用し、電源系機器に焦点をあてて故障の予兆の早期検出を試みた。

2. 対象データ

故障の予兆検知を行い、その有用性を検証するには、運用中の機器のセンシングデータが存在することや異常発生を表す情報が記録として残っていること等の条件がある。これらの条件を満たし、かつ、約 10 年間の運用により十分なデータが溜まっている X 線天文衛星「すざく」を故障の予兆検知の対象とした（表 1）。

表 1. X 線天文衛星「すざく」基本情報

打ち上げ	2005 年 7 月 10 日
科学観測終了	2015 年 8 月 26 日
軌道	高度 550km, 傾斜角 31 度の円軌道
軌道周期	約 96 分

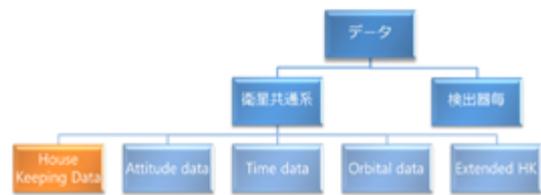


図 1. すざく衛星及び関連システムから得られるデータの構成

衛星から得られるデータには、図 1 のように HK (HouseKeeping) データや検出機器毎の観測データなどがあるが、それらのほぼ全てが DARTS [1]上で公開されている。また、すざくのホームページ [2]上では運用担当者による当時の運用ログ、観測ログを確認することも可能である。運用、観測のログからは SAFEHOLD (何らかの原因で異常が発生した場合、熱電氣的に安全な姿勢を保持するモード) が発生した原因を確認することができる。これらのログを調査することで、電源系機器の異常が原因と見られる SAFEHOLD を 7 箇所特定した。

運用期間中のバッテリー電圧と、電源系機器の異常が原因と考えられる SAFEHOLD のタイミングを示したものが図 2 である。打ち上げ後から 2012 年直前までは、バッテリーの電圧値が安定的に推移しているが、2012 年以後は、値が不安定になっていることが読み取れる。実際、太陽電池の発生電力が打ち上げから 7 年目に急激に低下したことがすざく衛星の観測運用終了に至る経緯の中に記されている [3]。そこには、観測運用終了に至った経緯としてバッテリーの劣化が一要因として挙げられている。よって、バッテリーの故障の予兆を検知することは宇宙機にとって重要なことである。

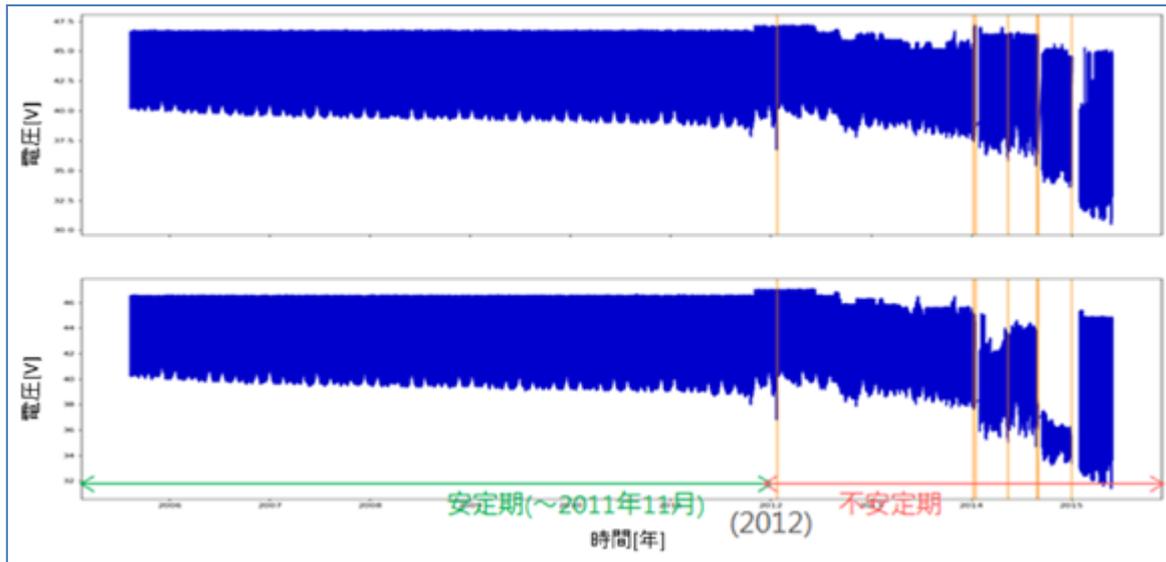


図 2. バッテリー電圧と電源起因の SAFEHOLD

(上：バッテリー A 電圧，下：バッテリー B 電圧，橙線が電源系機器起因の SAFEHOLD)

表 2. SAFEHOLD 発生時刻

No.	SAFEHOLD 時刻
1	2012-01-24 01:27:46
2	2014-01-09 18:39:19
3	2014-01-16 19:21:00
4	2014-05-15 20:26:00
5	2014-08-28 04:27:06
6	2014-09-04 00:11:21
7	2015-01-03 07:59:00

すざく衛星の運用中に発生した SAFEHOLD のうち、電源系機器の異常が原因と見られる SAFEHOLD を表 2 にまとめた。この計 7 回の SAFEHOLD の発生をできる限り早期に検出することを目標とする。

2011 年 12 月以降は、バッテリーの状態が常時不安定となっており、バッテリーの使用方法についても工夫がなされ、頻繁にモードが変更されていた。先行研究 [4]を参考に運用期間を安定期フェーズ（～2011 年 12 月）と不安定期フェーズ（2011 年 12 月～）の 2 つに分け、不安定期に入り始めたタイミングの SAFEHOLD を検出することを重視して確認を行う。

SAFEHOLD 発生の予兆を検出するための手法として機械学習を用いた。機械学習は、大量のデータの中に潜んでいる傾向を学習によって導き出し、得られた学習結果に新たなデータを投入して将来の予測をする手法である。学習はバッテリーの電圧や電流、太陽電池の温度など電源系機器に関わる様々な HK データを組み合わせで行った。評価は学習済みモデルを使用して予測を行った際の予測誤差を故障の予兆とみなして実施した。そのうち有用な結果が得られた 2 系統のバッテリー電圧・温度の計 4 項目のデータを利用した学習結果を以降に示す。

3. 異常検知手法

文献 [5] [6] [7]や先行研究 [8] [9] [10]を参考に今回の課題に適合する異常検知手法を選定して評価を行った。その中で、不安定期の最初の異常を検出可能であった 4 つの手法について概要を

示す。また、どの程度異常であるかを示す値（異常度）の算出方法についても合わせて示す。

その前に、各手法共通で実施するデータ前処理と学習データの抽出方法について説明する。

異常検知に用いるデータには欠損値が存在し、また、電圧と温度で出力されるタイミングが異なるため学習前にデータを整形する必要がある。そのため、事前に線形補間を行い、一定間隔で均一にリサンプリングしている。欠損の間隔が大きい場合はその期間のデータは対象外とした。

電源系機器は常時稼動しているため常に消耗している。10年という長期間にわたって連続稼動する過程で、正常時のデータの分布範囲も時間と共に変化していく。そのため、最初の期間の学習データを使用し続けていると、正常時でも徐々に異常度が高くなってしまふ。これを避けるため、各手法では、月が替わる毎に直近1月前のデータ（約60,000点）を訓練データとして学習し直している。例を図3に示す。なお、SAFEHOLDが発生した異常時のデータは学習から除外している。

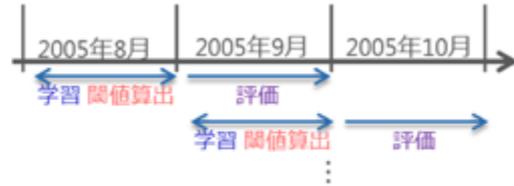


図 3. 学習・評価の算出例

3.1 kNN を用いた異常検知

kNN (K Nearest Neighbor, k 近傍法) は、k 個の近傍との距離をもとに近い点は同じグループに属するという考えからクラス分類に使用される手法である。今回は、最近傍のデータとの距離を測定するために用いている。また、kNN は傾向を学習するのではなく、1点1点との距離を測定するため、最近傍との距離の算出に時間がかかる手法である。

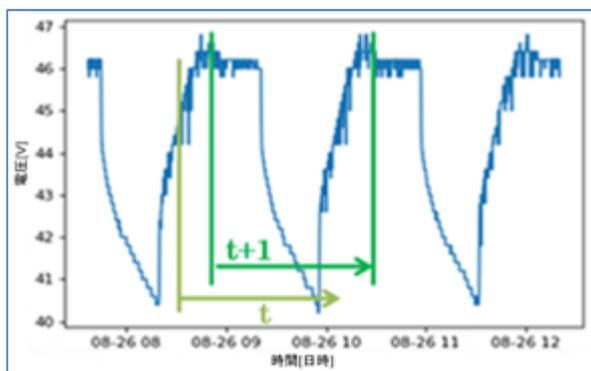


図 4. バッテリー電圧の周期

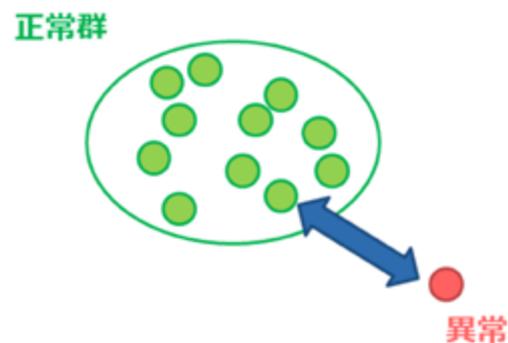


図 5. kNN 異常判定

バッテリーの電圧は衛星の軌道周期と同様の周期があり、約96分で1周期となる。横軸を時間、縦軸を電圧で表したものが図4である。この1周期を144次元(40秒刻み)のデータとして部分時系列を作成した。その部分時系列を学習し、最近傍の距離を測定すると、図5のように正常なデータは近くに密集し、異常なものは遠くに分布する。よって、最近傍までの距離を異常度とし、一定以上異常度が高いものを異常とみなす。

3.2 RNN を用いた異常検知

RNN (Recurrent Neural Networks) は、時系列データの扱いに特化した NN (Neural Network) の

一種である。NN は人間の脳の構造を模した機械学習のアルゴリズムであり、RNN は過去の隠れ層を取り入れることにより時系列データの予測を可能としている。RNN の構造を図 6 に示す。

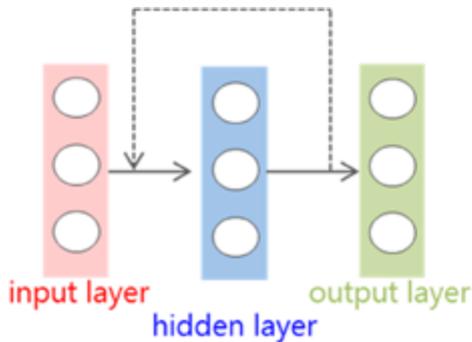


図 6. RNN モデル

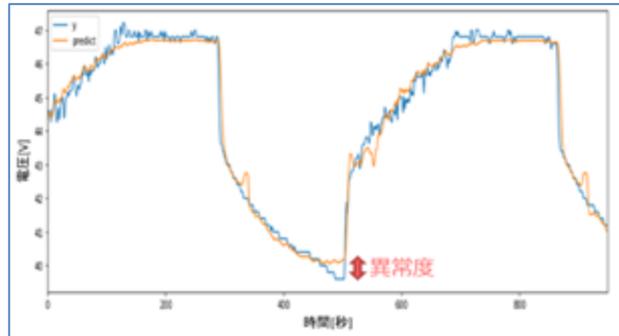


図 7. RNN 実際の値 (青) と予測値 (橙)

シーケンス長を 30、隠れ層のノード数を 400 とし、直近の部分時系列を用いて現在値を予測するように学習し、実際の値と予測した値を比較する。正常時のデータを学習したモデルでは、異常時のデータを予測できないため、予測値と実際の値の差が大きいほど異常であると考えられることができる。よって、図 7 のように予測結果と実際の値の差 (2 乗誤差) を異常度とし、異常を検出する。

3.3 AutoEncoder を用いた異常検知

AutoEncoder (自己符号化器) は、NN を用いて特徴表現を獲得する機械学習の手法である。入力層と出力層に同じデータを用いることで、ニューロン数を削減した中間層にデータの特徴を獲得することができる。AutoEncoder の構造を図 8 に示す。

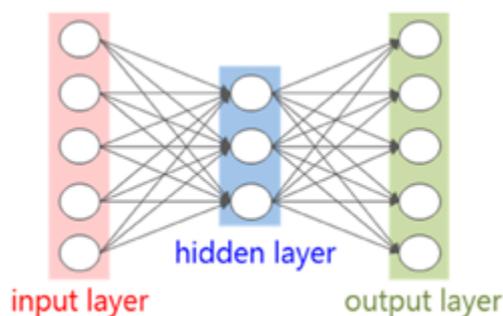


図 8. AutoEncoder モデル

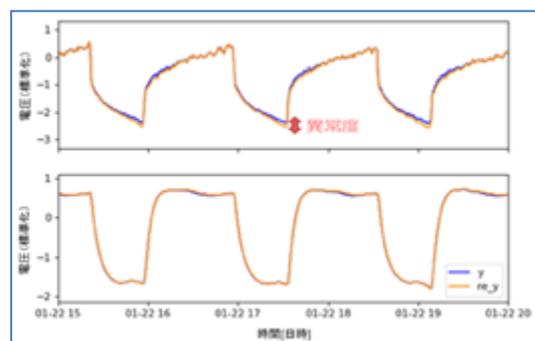


図 9. AutoEncoder 実際の値 (青) と再構成 (橙)

バッテリーの電圧と温度の計 4 次元のデータを入力し、項目間の関係を学習する。入力した学習データを出力層でそのまま再現するように学習した NN は、学習データと傾向が異なる異常時のデータを正しく再構成することができない。図 9 のように実際の値と再構成した結果の差（2 乗誤差）の合計を異常度とし、異常を検出する。

3.4 OneClassSVM を用いた異常検知

SVM（サポートベクターマシン）は、マージン最大化という手法を用いてデータを 2 クラスに分類する教師あり学習の手法である。OneClassSVM は、SVM と同様の手法を用い、データを 1 クラスに分類することで外れ値を検出する教師なし学習の手法である。データのまわりに境界をつくりデータがそのクラスに属するかを判定し、それ以外は外れ値とみなす。

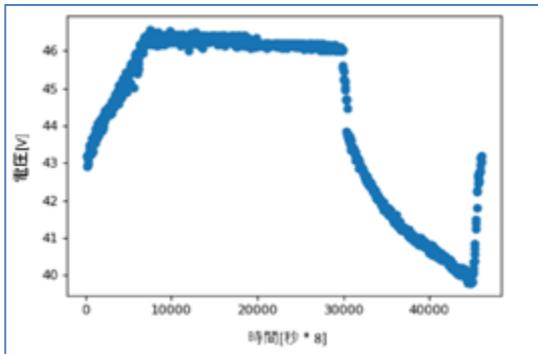


図 10. 周期の開始をあわせたバッテリー電圧

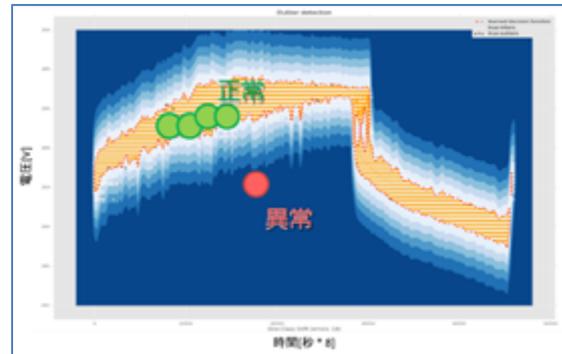


図 11. OneClassSVM 異常判定

図 10 のように、バッテリー電圧値の周期の開始タイミングを統一し、各タイミングの項目の値を学習させる。学習した結果、生成される境界を図 11 に示す。正常の場合は境界内にプロットされ、異常の場合は境界外の離れた位置に存在することになる。境界からの距離を異常度とし、異常を検出する。OneClassSVM での学習には RBF カーネルを用いた。

4. 異常検知結果・考察

4.1 異常検知結果

3 章で説明した 4 つの手法を用いて異常を検出した結果を図 12, 図 13, 図 14, 図 15 に示す。

正常状態と異常状態を区別するために閾値が必要であるが、訓練データが月毎に異なるため一定値を基準とすることはできない。そのため、訓練データに関して異常度を算出し、その異常度の標準偏差に一定値を掛け合わせた値を各月の閾値として算出した。閾値に掛け合わせる一定値は、安定期に誤報が起らないよう手法毎に調整している。図の縦軸は異常率（学習データの異常度/訓練データから算出した閾値）とし、「1」を超えたもの（閾値を超えたもの）を異常とみなす。

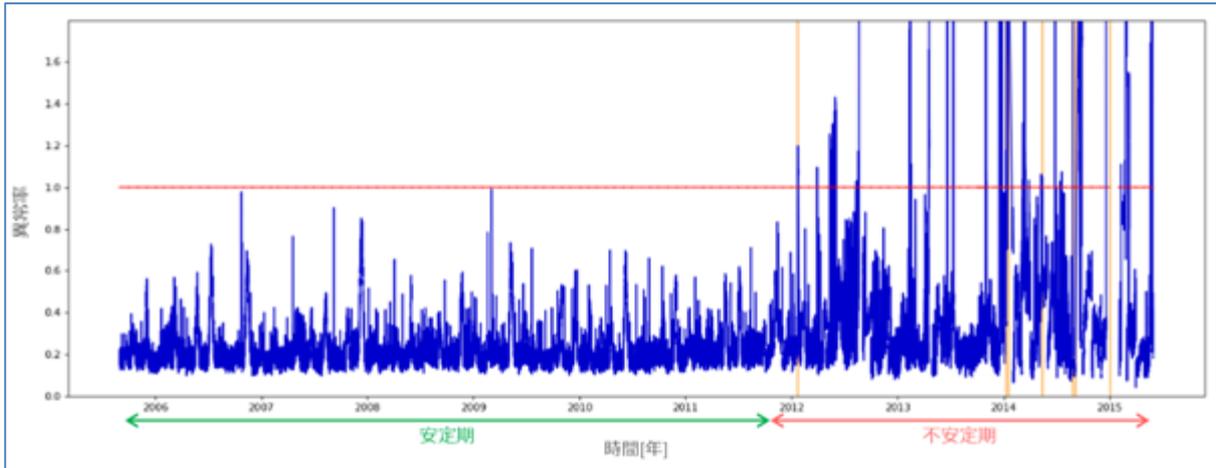


図 12. kNN 異常率 (赤線：閾値，橙線：対象の SAFEHOLD)

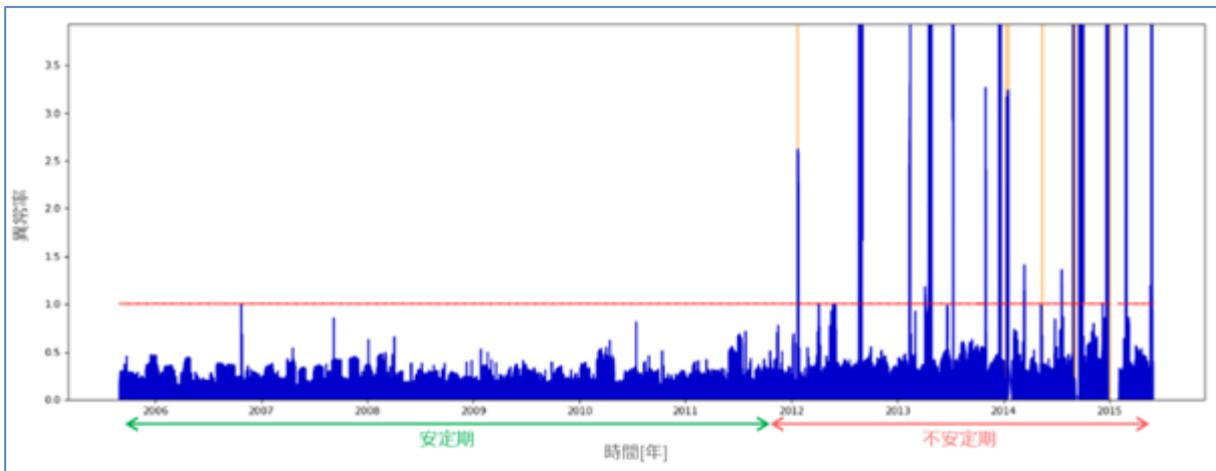


図 13. RNN 異常率 (赤線：閾値，橙線：対象の SAFEHOLD)

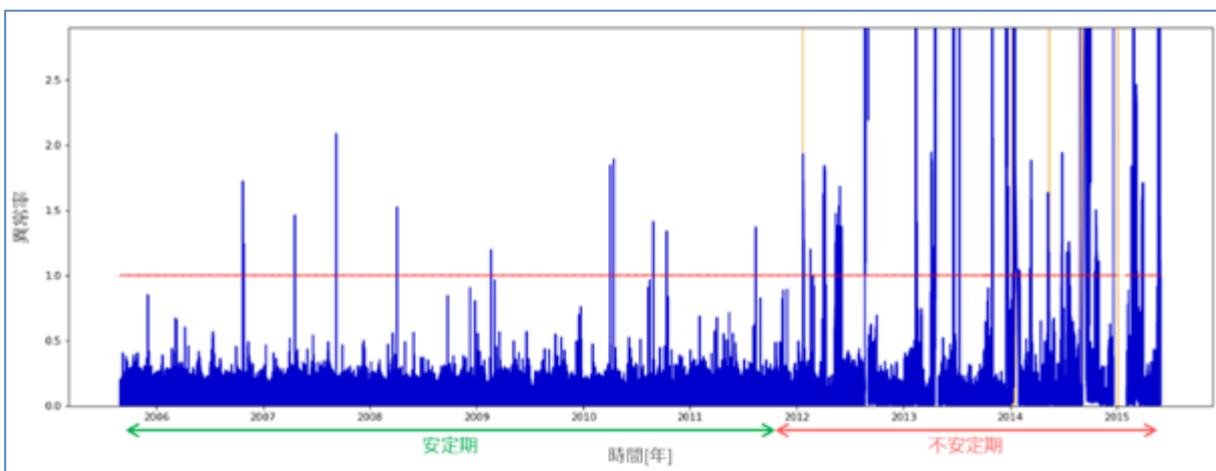


図 14. AutoEncoder 異常率 (赤線：閾値，橙線：対象の SAFEHOLD)

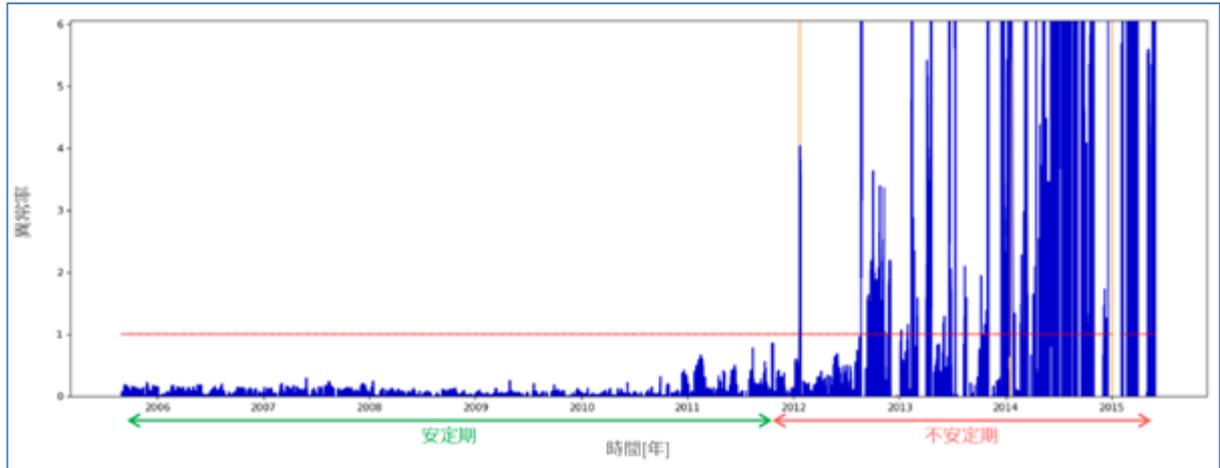


図 15. OneClassSVM 異常率 (赤線：閾値，橙線：対象の SAFEHOLD)

4.2 考察

各手法での検出可能時間 (SAFEHOLD をどれくらい前に検出することができたか, SAFEHOLD 前に異常度が閾値を超えた時間) を表 3 に示す。2 章で述べたとおり, 不安定期は, バッテリーの状態が不安定になり, モード変更も行われていたことから, 異常度も高く検出されている。ここでは, 安定期の誤報発生状況と不安定期初回の SAFEHOLD の検出結果に焦点を当て, 学習時間および異常度算出時間を含めて考察を行った。

表 3. 各手法での故障の予兆検知まとめ

	SAFEHOLD 検出可能時間 (時間前)							SAFE HOLD 検出数	安定 期誤 報数	学習時間 (月あたり)	異常度 算出時間 (月あたり)
	1	2	3	4	5	6	7				
kNN	13	42	115	×	×	×	328	4/7	0	○ (1分未満)	△ (10分未満)
RNN	28	×	110	×	×	×	327	3/7	0	△ (約30分)	○ (数秒)
AutoEncoder	12	21	117	×	×	×	328	4/7	11	△ (約10分)	○ (数秒)
OneClassSVM	44	108	117	126	×	140	328	6/7	0	○ (数秒)	○ (数秒)

表 4. kNN の閾値と誤報の関係

	SAFEHOLD 検出可能時間 (時間前)							SAFEHOLD 検出数	安定期 誤報数
	1	2	3	4	5	6	7		
kNN	13	42	115	×	×	×	328	4/7	0
	25	53	115	×	×	×	328	4/7	3
	36	75	115	125	×	×	328	5/7	76

各手法において不安定期の初回の SAFEHOLD を 12 時間以上前から検出可能であることを確認した。特に、RNN と OneClassSVM は安定期の誤報がなく、1 日以上前から検出することが可能である。閾値の調整次第では、誤報も増えるが、検出可能時間を長くすることが可能である。安定期の誤報を許容するのであれば、kNN も 1 日以上前から故障の予兆を検出することが可能である。例えば、表 4 のように 3 件の誤報を許容するのであれば、kNN も 1 日以上前から故障の予兆を検出することが可能である。

以上のことから、実際の宇宙機の運用においても、電源関連の値から算出された異常度やその傾向から故障が発生しそうであるという予兆を検出することが可能である。運用要件にあわせて閾値を調整することで、より早く故障の予兆を検出することもできる。

今回の結果では、すぎく衛星に対しては、安定期の誤報が少なく、より早期から多くの SAFEHOLD を検出することができ、かつ学習及び異常度算出にかかる時間も比較的短い OneClassSVM が一番有用な手法であるといえる。このような結果となった要因の 1 つとして、kNN や RNN, AutoEncoder はクラスタリングや予測値の推定に使用される手法であり、それらを応用して異常検知に用いたのに対し、OneClassSVM は外れ値を検出することを目的とした手法であり、今回の異常検知問題に関しても、その手法の特性が有効に働いたことが考えられる。

5. まとめ

機械学習技術を用いて、運用データから故障を事前に検知するためのモデル構築を行った。様々なアルゴリズムを検証した結果、初回の故障の予兆検出に着目すると kNN, RNN, AutoEncoder, OneClassSVM の 4 つのモデルで良好な結果が得られた。また、安定期の誤報数や検出可能時間から OneClassSVM が 4 手法の中でも一番良い結果が得られた。安定期フェーズでの誤報を限りなく抑えつつ、不安定期フェーズの中でも比較的安定した期間の最初の異常を 1 日以上前から検出ことが可能であり、実運用上有用であると考えられる。

今後は他の衛星へも同様の手法を適用できるか確認し、現在運用中の宇宙機や将来打ち上がる宇宙機へ適用できるかを検討する。また運用で使用するためには、運用担当者にとって宇宙機の状況が分かりやすいことも重要であるため、異常度が高くなると警告が発生するようなシステムのプロトタイプを作成し、宇宙機の運用に貢献することを目標とする。

参考文献

- [1] C-SODA at ISAS/ JAXA, “DARTS,” [オンライン]. Available: <http://www.darts.isas.jaxa.jp>. [アクセス日: 21 8 2018].
- [2] JAXA 宇宙科学研究本部 高エネルギー天文学研究系, “すぎくホームページ,” [オンライン]. Available: <http://www.astro.isas.jaxa.jp/suzaku/>. [アクセス日: 29 8 2018].
- [3] 常田佐久, *X線天文衛星「すぎく」の科学的成果について*, 2015.
- [4] 前田良知, “バテてもめげない BAT サッパリわからん SAP,” *天文月報*, pp. 14-20, 1 2016.
- [5] 井手剛, 入門 機械学習による異常検知—Rによる実践ガイド, コロナ社, 2015.
- [6] 井手剛, 杉山将, 異常検知と変化検知, 講談社, 2015.
- [7] 巢籠悠輔, 詳解 ディープラーニング ~TensorFlow・Kerasによる時系列データ処理~, マイナビ出版, 2017.
- [8] 乾稔, 矢入健久, 河原吉伸, 町田和雄, “次元削減の再構成誤差を用いた異常検知手法の比較,” 2009.
- [9] 矢入健久, 乾稔, 河原吉伸, 高田昇, “次元削減とクラスタリングによる宇宙機テレメトリ監視法,” 2011.
- [10] 櫻田麻由, 矢入健久, “オートエンコーダを用いた次元削減による宇宙機の異常検知,” 2014.

背面投影型デジタル地球儀 Dagik Earth のための 半球面マルチタッチパネルの開発

小山 幸伸^{*1}, 廣田 尚樹^{*1}, 増田 花乃^{*2}, Pipatpol TANAVONGCHINDA^{*3}, 佐藤 弘^{*4}

Development of the hemispherical multi-touch panel for the rear-projection digital globe “Dagik Earth”

Yukinobu KOYAMA^{*1}, Naoki HIROTA^{*1}, Hanano MASUDA^{*2}
Pipatpol TANAVONGCHINDA^{*3}, Hiro SATO^{*4}

Abstract

The digital globe “Dagik Earth” is an effective tool for general citizens to study global environments. Although various user-interfaces have been tested for Dagik Earth, such a user-interface that can control the contents by directly touching the digital globe itself has not been realized. Therefore, we started development of a hemispherical multi-touch panel for the rear-projection digital globe by adopting the FTIR technology. We built a hemispherical multi-touch panel system using infrared LED, infrared Web camera, CCV, and TUIO.

While the sensitivity is insufficient for bright environment, we were successful to realize the hemispherical multi-touch panel under the limited condition that all the external lights are blocked except for the projector and the infrared LED.

Keywords : Dagik Earth, Hemispherical Multi-touch Panel, Infrared LED, Infrared Camera

概要

地球環境を市民が学習するための道具として、デジタル地球儀 Dagik Earth が有効と考える。これまで様々なユーザーインターフェースが Dagik Earth に試行されてきた。しかしながら、デジタル地球儀そのものに触れてコンテンツを操作するユーザーインターフェースは実現していなかった。そこで我々は、FTIR方式を用いた背面投影型デジタル地球儀 Dagik Earth のための半球面マルチタッチパネルの開発に着手した。赤外線 LED、赤外線 Web カメラ、CCV、TUIO を用いて半球面マルチタッチパネル・システムを構築した。未だ感度は十分ではないものの、プロジェクタおよび赤外 LED を除く外部の光を遮った環境下において、半球面マルチタッチパネルを実現した。

キーワード ダジックアース, 半球面マルチタッチパネル, 赤外 LED, 赤外線カメラ

doi: 10.20637/JAXA-RR-18-008/0003

* 平成 30 年 12 月 19 日受付 (Received December 19, 2018)

^{*1} 大分工業高等専門学校情報工学科 (Department of Information Engineering, National Institute of Technology, Oita College)

^{*2} 福井大学工学部情報・メディア工学科 (Department of Information Science, Faculty of Engineering, Fukui University)

^{*3} カセサート大学計算機工学部 (Department of Computer Engineering, Kasetsart University)

^{*4} 熊本大学工学部情報電気電子工学科 (Department of Computer Science and Electrical Engineering, Faculty of Engineering, Kumamoto University)

1. 背景および目的

西日本に甚大な被害をもたらした平成30年7月豪雨に代表される度重なる災害によって、専門家のみならず子供から大人に至る一般市民の地球環境に関する注目度は高まっている。市民がそれを身近に学習するための道具として、デジタル地球儀が有効と考える。JVC ケンウッドの触れる地球¹⁾や Global Imagination の The Magic Planet²⁾など、多人数が一度に観覧できるデジタル地球儀が市販されているが、小規模な組織で導入するにはいずれも高価である。他方で、科学館や小中学校において導入可能な低予算で実現可能なデジタル地球儀として Dagik Earth³⁾⁴⁾が挙げられる。これまでキーボード、マウス、トラックボール、Wii リモコン、バランス Wii ボード、Kinect、Leap Motion、赤外線リモコン、QR コードリーダーなどに代表される様々なコントローラーが、Dagik Earth に試行されてきた。しかしながら、デジタル地球儀そのものに触れてコンテンツを操作するユーザインターフェースは実現していなかった。そこで我々は、背面投影型デジタル地球儀 Dagik Earth のための半球面マルチタッチパネルの開発に着手した。

2. タッチパネルの仕組みと原理

2.1 光波におけるスネルの法則

図 1 に半球面マルチタッチパネルの概念図を示す。赤外 LED によってアクリル半球の断面から入射した赤外線は、アクリルと空気と境界面における全反射を繰り返して反対側の断面に到達する。この途中経路に触れることによって、赤外線は指に反射する。これを赤外線カメラによって検出し、PC のポインティングデバイス制御と結び付けることによってタッチパネルを実現する。このようなタッチパネル方式は Frustrated Total Internal Reflection (FTIR)方式と呼ばれる。次に前出した赤外線の全反射について考える。接合した2つの媒質 A および B の屈折率を n_A および n_B とし、その入射角および屈折角を θ_A および θ_B とすると、スネルの法則 $n_A \sin \theta_A = n_B \sin \theta_B$ が成り立つ。図 2 より得られるアクリルの屈折率を $n_A \approx 1.49$ 、そして空気の屈折率を $n_B = 1.00$ として代入すると、 $\sin \theta_A = (n_B/n_A) \sin \theta_B = (1.00/1.49) \sin(\pi/2)$ となり、臨界角 $\theta_A \approx 38.5^\circ$ を超えた場合に全反射が起こる。

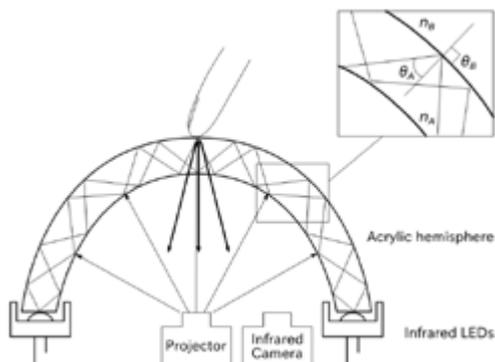


図 1 半球面マルチタッチパネルの概念図。

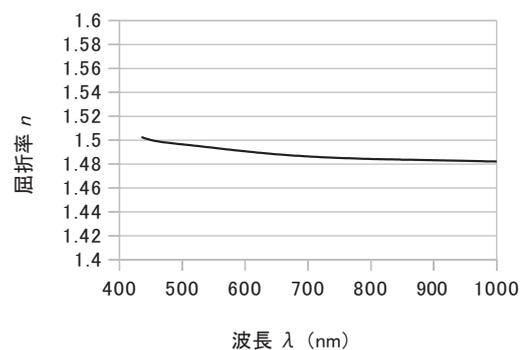


図 2 光の波長に対するアクリルの屈折率⁶⁾。

2.2 赤外 LED とブラケット

赤外線のパーク波長が940nm, 照射角が30°, 順方向電流100mA 時の放射強度が40mW/srである OptoSupply 社の直径3mmの高輝度赤外 LED の OS15LA3131A[®]を使用した.図 3にこの赤外 LED 駆動回路を示す.試験段階の現在は, 順方向電流を100mA ちょうどにするために, 安定化電源を用いて11.6V を供給している. 直径40cm 厚さ3mm のアクリル半球の断面を240番から2000番の耐水やすりで磨いた後に, アクリル研磨剤⁷⁾を用いて滑らかにした.このアクリル半球の断面に対して赤外 LED 光の中心軸を垂直に保持するために, ダブルクリップによって作成した LED ブラケットによってアクリル板を挟み込んで固定した. この LED ブラケットは, PLUS 社の幅19mm のエアかるダブルクリップ CP-103AK を3mm の穴あけ加工することによって作成した. LED ブラケットに搭載した赤外 LED をアクリル半球の断面に8個取り付け, その設置個所以外は反射させるために白色塗料を塗った.

これまで様々な背面投影型 Dagik Earth の実装が公開されてきたが, それらの多くは, アクリル半球を支持するためのフランジ付きのアクリル半球が利用されてきた.しかしながら, 本取り組みではアクリル板断面から赤外線を入射させるために, 荷重を支えるためのフランジを付けることができない. そこで図 4のアクリル半球の支持具を作成し, 図 5のようにアクリル半球を4点支持した.

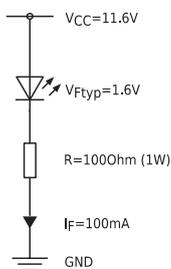


図3 赤外LED 駆動回路.

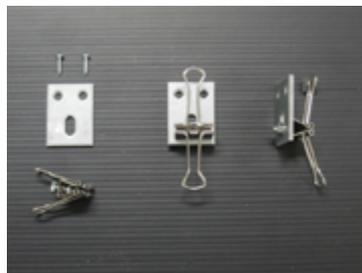


図4 ダブルクリップを用いた着脱式のアクリル半球の支持具.

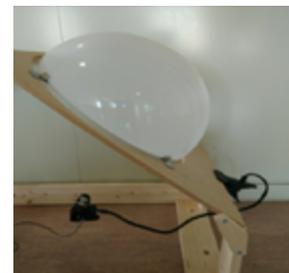


図5 フランジ無しアクリル半球の4点支持, およびフレキシブルアームを用いた赤外線 Web カメラの支持を示す.

2.3 赤外線 Web カメラ

Dagik Earth と同様に, マルチタッチパネル機能も安価に実現することが望ましい.そこで本研究では, 高価な既製品の赤外線カメラを使わず, 手頃な価格の Web カメラを, 赤外線 Web カメラに改造した. 使用した Web カメラは, バッファローの BSWHD06M である.多くの Web カメラと同様に, 図 6の左図のような赤外域をカットするフィルターが組込まれていたもので, ピンセットを用いてこれを除去した. さらには, 図 7の様に Web カメラの CMOS センサ前に, 光吸収・赤外線透過フィルター (Infra Red (IR) フィルター) FUJIFILM IR84を組込んだ. 短波長の可視光が IR フィルターによって吸収される時の分光感度の概念図が図 6の右図である. 以上に加えて, アクリル半球のエッジにおいて反射した赤外線をカメラが検出ないように, 3φ のアルミワッシャーを CMOS センサー前に組込むことによってマスクした. このマスクの有無による検出画像の差異を図 8に示す.

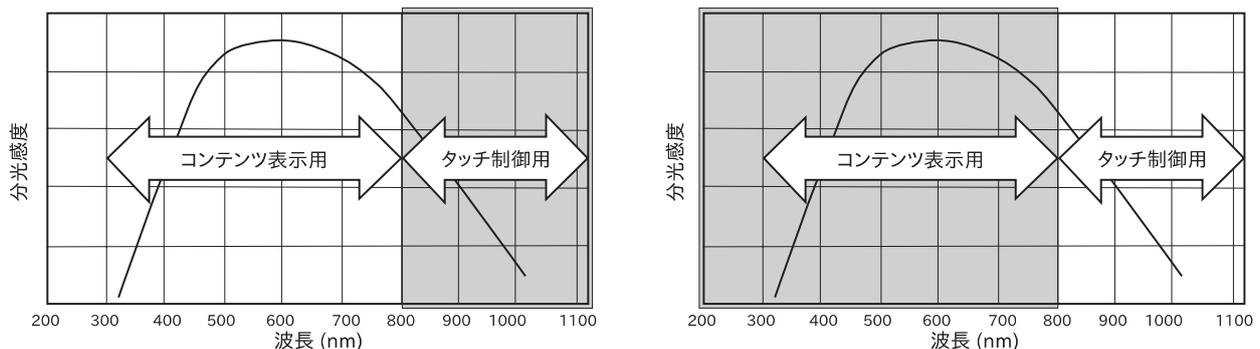


図6 Web カメラの分光感度曲線の概念図を示す. 多くの Web カメラは, 左図のように赤外域をカットするフィルターが組み込まれている. このフィルターを除去した後に, 可視光除去フィルターを組込んで右図のような分光感度特性にする.



図7 適切なサイズに切断した IR フィルターを，CMOS センサ前に組み込む様子を示す．マスク用の 3φ のワッシャーも同様に組み込む．

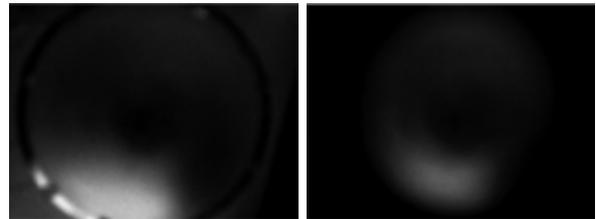


図8 左図は，アクリル半球のエッジで反射した赤外線が見える．右図は，マスクを用いた際の赤外線 Web カメラの画像である．

2.4 タッチ箇所の検出とポインティングデバイス制御の紐づけ

指が触れた箇所，すなわち赤外線の反射箇所を検出するために，フリーのライブラリである Community Core Vision(CCV)⁸ Version 1.5の付属アプリケーションを利用した．図9にCCVによるマルチタッチ検出の様子を示す．左上の画像が赤外線 Web カメラによる入力画像である．ただし，入力画像に加えて，タッチ箇所の重心も描画されている．下の4つの画像は，背景除去，ノイズ除去，ハイパスフィルタ，弱いピクセルの増幅という各々のフィルターを入力画像に適用した画像である．右上の画像は全フィルターを適用後の画像であり，指が半球に触れることによって作られる白い小塊の1つ1つにユニークな識別子が付与された．さらには，小塊の現在座標が3333番ポートから出力されることを確認した．仮に小塊が移動しても，ひとたび付与された識別子はそのままであり，時々刻々と小塊の座標が出力される．しかしながら，小塊がひとたび無くなった後に再び現れたとき，すなわち指をひとたび離してまた同じ指で触れた場合においては，新たな識別子が付与される．

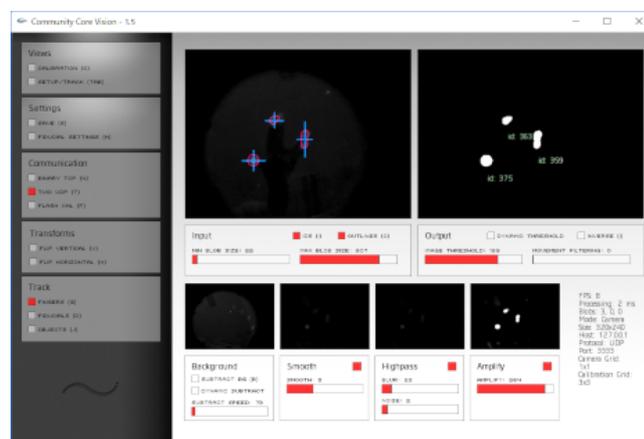


図9 CCVによるマルチタッチ検出の様子を示す．



図10 遮光テスト環境．



図11 半球面マルチタッチパネルを操作している様子を示す．左図が回転前で，右図が回転後である．

次に3333番ポートから出力される各々の小塊の現在座標をマルチタッチイベントに紐づけるために、Tangible User Interface Object (TUIO)⁹⁾を用いた。TUIOにはサンプルアプリケーションが用意されていたものの、我々が必要とするキーボード処理の機能不足があったため、Tuio.jarライブラリを利用しつつ追加実装したDagikTouchController.jarを開発した¹⁰⁾。小塊の座標をmouseMoveメソッドに与える実装を行っている。

CCV、およびTUIOを用いたDagikTouchControllerは共にJava言語で書かれているため、Java Virtual Machineが動作する環境であれば動作する。さらに赤外線 Web カメラは、USB インターフェースを持つ PC に接続される。そこで我々は、Windows 10, Mac OS X, Raspbian 環境下において本システムの動作を確認した。

3. 実験および考察

図 10に遮光テスト環境を示す。Dagik Earth 用コンテンツを投影するためのプロジェクタの光路長を確保するため、高さは1.98mである。左図のように、19×89mmの木材をコーススレッドによって固定した骨組みに、リベットを付けた黒いブラダンロープでもやい結びして固定した。前面は、右図のように2重にしたサテン生地によって遮光している。遮光しつつ前面から内部に片手を入れてタッチ操作できるように設計されているため、図 11のとおり遮光テスト環境下において半球面マルチタッチパネルの操作テストを行った。左図のように天頂に触れた後に、右図の状態まで手を触れたまま動かすことに従って、コンテンツが回転することを確認した。複数の赤外 LED が強め合う箇所であつ、カメラの視野の中心であることから、天頂付近の感度が良かった。球の端の感度を改善するには、赤外線 Web カメラを2×2の4台用いて放射状に設置し、半球の端に至るまでカメラの視野を網羅すれば良いと考える。幸いなことに CCV は複数台のカメラ入力に対応しているため、半球の端の感度向上は実現可能と考える。

アクリル半球の支持も問題なく、天頂付近はそれほど強い力で押すことなくタッチ検出が可能であつた。しかしながら、コンテンツの回転動作時には横向きの動きに意識が取られ、球面に対して垂直な方向の力が弱まるため、回転動作の反応が良くなかつた。この押圧を定量的に測るために、インターリンク社の圧力センサーFSR402¹¹⁾を手袋の指先に付け、Arduino Uno の10ビット AD コンバーターによって、計測する仕組みを作成した。様々な被験者によるコンテンツ回転時の押圧測定などの定量的な評価は、今後の課題である。

4. まとめ

本論文では、FTIR方式を用いた背面投影型デジタル地球儀Dagik Earthのための半球面マルチタッチパネルの開発状況を報告した。

赤外線 LED、赤外線 Web カメラ、CCV、TUIOを用いて半球面マルチタッチパネル・システムを構築した。未だ感度は十分ではないものの、プロジェクタおよび赤外 LED を除く外部の光を遮った環境下において、半球面マルチタッチパネルを実現した。今後は、カメラを複数台を使うなど、タッチパネルの応答性向上を狙う。本研究は、遮光テスト環境下において検証を行ってきたため、実際に近い環境下で試験と改良を重ね、アウトリーチ現場へ提供することが次なる課題である。

謝辞

Dagik Earth 機材一式の貸し出し、および助言を頂いた、京都大学大学院理学研究科の齊藤昭則准教授、小田木洋子技術補佐員に、著者一同は感謝する。

参考文献

- 1) 触れる地球, <http://www3.jvckenwood.com/t-earth80/jp/>, 参照日: 2018年11月29日.
- 2) The Magic Planet, <http://globalimagination.com/>, 参照日: 2018年11月29日.
- 3) Dagik Earth, <https://www.dagik.net/>, 参照日: 2018年11月29日.

- 4) 齊藤 昭則, 津川 卓也, 市川 浩樹, 島田 卓也, 多様な環境においてデジタル立体地球儀を実現するための ダジック・アースの開発, 宇宙科学情報解析論文誌, 第6号, 2017, doi:10.20637/JAXA-RR-16-007/0012.
- 5) アクリル, アクリレート, ルーサイト, パースペクス, プレキシガラス屈折率, <https://www.filmetricsinc.jp/refractive-index-database/Acrylic/>アクリル-アクリレート-ルーサイト-パースペクス-プレキシガラス, 参照日 : 2018年11月29日.
- 6) OSI5LA3131A, <http://www.tme.eu/en/Document/c502fbb3d9b8c4061b369c7ceb587334/OSI5LA3131A.pdf>, 参照日 : 2018年11月29日.
- 7) 加工溶剤・用品, <http://www.acrysunday.co.jp/products/article/>, 参照日 : 2018年11月29日.
- 8) Communiti Core Vision, <http://cev.nuigroup.com/>, 参照日 : 2018年11月29日.
- 9) TUIO implementations, <http://www.tuio.org/?software>, 参照日 : 2018年11月29日.
- 10) DagikTouchController.java, <https://gist.github.com/koyamalmsteen/acdbd9a1cfa3cef1f2bdec259739a30f>, 参照日 : 2018年11月29日.
- 11) Force Sensing Resistors, FSR402, Interlink Electronics, <http://akizukidenshi.com/download/ds/interlinkelec/fsr.pdf>, 参照日 : 2018年11月29日.

「はやぶさ2」ハードウェアシミュレータに係る レイトレーシングソフトウェアの開発と評価

三浦 昭^{*1}, 武井 悠人^{*1}, 山口 智宏^{*1,*2}, 高橋 忠輝^{*1}, 佐伯 孝尚^{*1}

Development and Evaluation of the Ray-Tracing Software Used in the Hardware-in-the-loop Simulator of Hayabusa 2

Akira Miura^{*1}, Yuto Takei^{*1}, Tomohiro Yamaguchi^{*1,*2}, Tadateru Takahashi^{*1}, Takanao Saiki^{*1}

概要

2014年12月に打ち上げられた小惑星探査機「はやぶさ2」は、2018年の夏から小惑星リュウグウ(162173, 1999 JU₃)の探査を続けている。係る探査においては、探査前に情報が得られないような状況下で、様々なクリティカル運用が予定されており、係る事前訓練や検証等が欠かせないものとなった。同様に関連する科学者も実際の観測に先立って訓練を積む必要があった。来たる実運用に向けて、信頼性を上げるための一助としてハードウェアシミュレータが開発された。その構成要素の一つとして、画像生成装置が開発され、「はやぶさ2」搭載の様々な光学機器の模擬データを生成することとなった。本稿においては、光学機器模擬の概要を記すと共に、そこで使われるレイトレーシング機能について、性能評価の概要を述べる。

Abstract

The Hayabusa 2 asteroid explorer was launched in December 2014 and has been exploring Ryugu (162173, 1999 JU₃) since the summer of 2018. The mission includes various critical operations under previously unknown circumstances. Also, scientists need to be trained in advance for actual observations. To increase the reliability of forthcoming operations, a hardware-in-the-loop simulator has been developed. One of the components of the simulator is the asteroid image generator, which generates various simulated data of optical instruments on board Hayabusa 2. In this paper, we describe the overview of the optical instruments simulation and the performance of the ray-tracing software used therein.

Keywords: Hayabusa 2, HIL simulator, Ray tracing

1 まえがき

本稿においては、「はやぶさ2」における各種訓練のために用意されたハードウェアシミュレータのコンポーネントである画像生成装置について、その中核をなすレイトレーシングソフトウェア開発の概要と性能評価を中心に述べる。

小惑星探査機「はやぶさ」に続いて開発された「はやぶさ2」は、C型小惑星のサンプルリターンミッションを担って2014年12月3日に打ち上げられた。「はやぶさ2」は2018年6月に小惑星リュウグウ(162173, 1999 JU₃)に到着し、2019年末までリュウグウに滞在する予定となっている。

小惑星リュウグウについては、事前に把握できた情報が少なく、小惑星到着に備えた検討準備が不可欠であった。そのような中で、実際の探査を想定した様々な訓練が計画され、実施されてきた。その代表的なものにRIO (Real-time Integration Operation) 訓練 [1] と LSS

(Landing Site Selection) 訓練 [2] が挙げられる。

LSS 訓練は「はやぶさ2」のタッチダウン地点選定のための訓練であり、リュウゴイドと呼ばれる仮想小惑星 [3] を用いて本番さながらの各種観測等を模擬した訓練が2017年度初頭から実施された [4][5][6][7][8][9]。訓練に供された画像の例を図1に示す。図中に付されている名称等は、模擬観測に基づいて理学研究者が命名したものである。

RIO 訓練においては、2017年度後半から2018年度初頭にかけて、実際の管制室も用いて様々な条件下でのリアルタイム訓練が繰り返し実施された。その概念図を図2に示す。

それらの訓練を支援するシステムのひとつが、ハードウェアシミュレータ (Hardware-in-the-Loop Simulator, HIL) である [11]。その概略を図3に示す。HIL シミュレータはRIO 訓練やトラブルシューティング、運用手順の検証等の用途を想定して開発されたものであり、探査

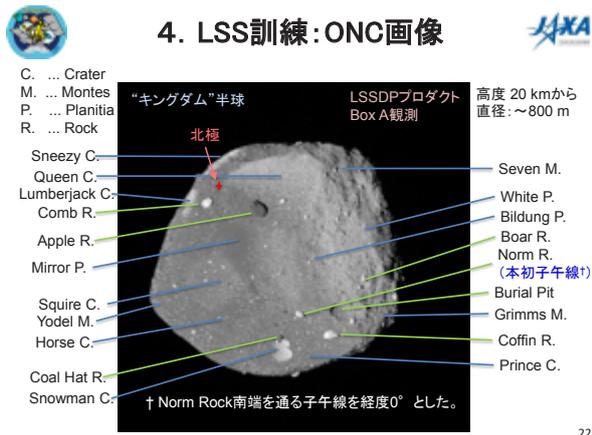


図 1: LSS 訓練における ONC 画像の例 (引用元: [10])

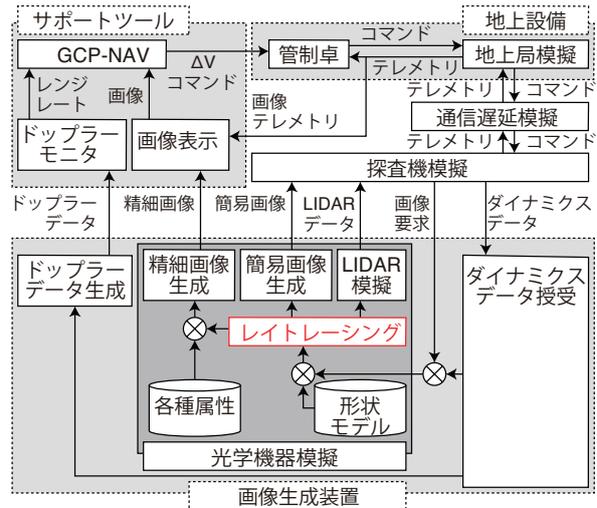


図 3: HIL シミュレータの概要と光学機器模擬 (OIS) の位置付け

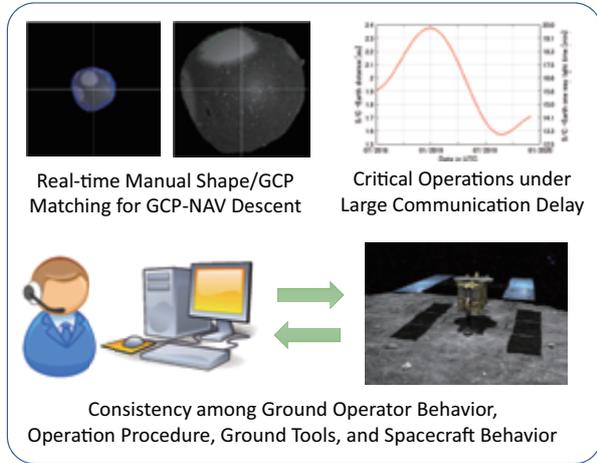


図 2: RIO 訓練概略 (引用元: [11])

機、通信環境、地上局等、様々な模擬コンポーネントで構成されている。そのひとつとして、小惑星探査に係る画像生成を主目的とした装置 (画像生成装置) が開発されている。画像生成装置においては、探査機を模擬するコンポーネントから受け取ったダイナミクスデータや、模擬観測に使用される光学機器の構成等に基づいて、画像やドップラーデータ等の模擬データが生成され、関係するコンポーネントに生成結果が提供される。

画像生成装置は RIO 訓練のみならず、LSS 訓練やその他の様々な訓練・検証のための画像生成にも供されており、本稿の執筆時点では、リュウグウの形状モデルに基づいた模擬画像生成等、さらに適用範囲を広げている。その画像生成装置において中核的な機能を提供しているのが、光学機器模擬 (Optical Instruments Simulator, OIS) である。

以下、次節で OIS 構築の概要を述べた後、その中でも共通要素として用いられ、かつ演算負荷の高いレイトレーシング機能に焦点を絞って、その検討・評価を行った結果について述べる。

2 OIS

本節では、図 3 中に示される光学機器模擬 (OIS) 構築の概要を述べる。

2.1 OIS が満たすべき要件等

2.1.1 速度・精度

各種訓練の成果を高めるためには、OIS は要求される応答時間内に訓練遂行に十分な品質の模擬画像を生成する必要がある。

例えば RIO 訓練で許容される遅延時間は、精細画像で 10 分程度、簡易画像で 1 秒程度となる。他方、LSS 訓練においては、RIO 訓練と比べると高い精度のレンダリング結果を提供する必要がある。例えばオーバーサンプリングを考慮すると、数万 × 数万画素程度のレンダリングが必要となる場合があり、画素毎の位置精度も相応のものが必要となる。例えば理学観測に用いる望遠カメラ (ONC-T) [12] の画角は 6 度程度であるから、このような場合、1 画素あたりの分解能は 1 秒角未満となる。

2.1.2 インタフェース

OIS ソフトウェアと他のプロセスとのインタフェースは以下の通りである。なおコマンドやデータ格納形式等を定めたインタフェース基準書は非公開であるため、ここでは概要を述べるに留める。

- 他プロセスとのインタフェースは TCP のソケット通信を用いる。

- コマンドは当該ソケット通信により受け取り、実行にあたってのステータスはソケット通信の返信として要求元に通知する。
- コマンドのフォーマットは JSON 形式とし、係るパラメータは事前に定義された形式で受け取るものとする。もしくは JSON 形式でパラメータを格納したファイルのパスをコマンドとして受け取る。
- 実行結果となるデータ群は前述のコマンドにより指定された形式で、指定されたファイルシステムに書き出す。

2.1.3 OIS に提供されるデータ等

OIS には、以下に述べるような形状モデルや各種属性等を取めたファイル群が提供される。

形状モデル 小惑星や探査機、その他模擬に必要な各種形状モデルが提供される。小惑星モデルは、その用途によって約 30 万ポリゴンから 4 億ポリゴン程度の形状モデルが用意される。

各種属性 以下のような属性を表すデータ群で構成されている。これらのデータは関連する理学関係者や搭載機器関係者から提供されるものであり、訓練の目的等に応じて画像生成に供される。

- 小惑星形状モデルの各ポリゴンに対して、地質等の属性を与える対応表。シミュレーションの用途によっては、他の理学データやボルダー識別等の対応表も提供される場合がある。
- 太陽やフラッシュランプ等の光源の特性。
- 各種地質属性（光学的特性等）。
- 搭載カメラ（フィルタ、CCD）等の特性。
- その他の補助的データベース。

画素値計算ライブラリ OIS ソフトウェアが画素値を計算するにあたって必要となる関数群が収められている。

2.2 OIS ソフトウェア

OIS ソフトウェアは、前述の要件を満たすために設計・構築された、レイトレーシングを基本としたレンダリングソフトウェアであり、各種カメラや LIDAR 等の光学機器を模擬し、画像等の模擬データを生成するものである。

なお実際の画像生成装置では用途毎に複数のプロセスが並行稼働することがあるが、図 3 中では便宜的に単一

の編成として表現している。例えば RIO 訓練を実施する場合、OIS としては精細画像生成を担うプロセスと簡易画像生成・LIDAR 模擬を担うプロセスの 2 つが並行稼働する。その他の関連するプロセスも同時稼働する場合がある。

2.2.1 処理の流れ

OIS ソフトウェアの処理について、その概略を以下に示す。

OIS ソフトウェアの起動 OIS ソフトウェアは、必要とされる形状モデルや各種属性情報等のデータを起動時に読み込み、以後の画像生成要求を受信するための TCP ソケットを生成する。係るポート番号はファイルに保存して他のプロセス等から参照可能な状態とする。

OIS ソフトウェアは原則として HIL シミュレータ内で各種コンポーネントと連携するものであるため、GUI 等のような対人インタフェースは持たない。その代わりに、TCP 通信をラップした、簡易のクライアント用 CLI を提供する。

画像生成要求、ダイナミクスデータの受信 OIS ソフトウェアは、画像生成の要求が発生する都度、その時点のダイナミクスデータと画像生成条件を TCP ソケット経由で受信する。LIDAR は現実には画像を生成するカメラではないが、ソフトウェア上では 1 ピクセルの深度（距離）画像として扱っている。画像生成に必要なとされるダイナミクスデータは、OIS ソフトウェア外のプロセスが担っており、その生成の詳細については本稿の対象外とする。

レンダリング OIS ソフトウェアは、受信したダイナミクスデータ等と、起動時に読み込まれた形状モデル等に基づいて画素値や距離情報を計算する。コマンド要求の内容によっては、予め提供されている各種属性や画素値計算ライブラリ等が使用される。

結果の保存 OIS ソフトウェアは、レンダリング結果の画素値や距離情報等を指定されたフォーマットでファイルに保存し、そのステータスを TCP ソケット経由で返信する。

待機 OIS ソフトウェアは、次の画像生成要求があるまで待機する。

2.2.2 設定ファイル

前述のように、OIS ソフトウェアでは JSON 形式でコマンドを得ることとなっているため、起動時に用いる各

```
{
  "proc_out_directory": "out",
  "proc_image_name": "image.pgm",
  "proc_render_image_width": 2048,
  "proc_render_image_height": 2048,
  "proc_out_image_width": 512,
  "proc_out_image_height": 512,
  "proc_sensor_id": 9,
  "proc_fast_simulation": 1,
  ...
}
```

図 4: JSON の書式例

種設定ファイルも原則として JSON で記述することとする。図 4 に JSON の書式例を示す。この例に記述されたパラメータは、起動時にも実行時のコマンドとしても使用可能なものである。

ここに例示された内容は、次のように解釈される。出力ディレクトリ (proc_out_directory) を “out” とし、ファイル名 (proc_image_name) を “image.pgm” とする。レンダリング幅 (proc_render_image_width) を 2048 とし、高さ (proc_render_image_height) を 2048 とする。出力幅 (proc_out_image_width) を 512 とし、高さ (proc_out_image_height) を 512 とする。すなわち、この例は 4 倍のオーバーサンプリングを要求している。用いるカメラとフィルタの組み合わせ (proc_sensor_id) は 9 番とし、地質属性等に依らない高速のレンダリング (proc_fast_simulation) を要求する。

形状モデル等、読込に時間を要する一方で訓練に係る一連のレンダリングにおいて変更する必要がないデータや、変更することが望ましくないパラメータ類は、プロセスの起動時に与えられた値のみを有効とする。そのようなパラメータの例を図 5 に示す。

ここに例示された内容は、次のように解釈される。プロセス ID (init_pid_file) を “run/pid.txt” に格納する。コマンド受信に使用する TCP ソケットのポート番号 (init_port_file) は “run/port.txt” に格納する。CCD 等の特性データベース (init_ccd_db) を “data/ccd” から読み込む。探査機の形状モデル (init_model_probe) を “models/probe.stl” から読み込む。小惑星の形状モデル (init_model_asteroid) を “models/ast.stl” から読み込む。小惑星の地質情報 (init_geological_unit_id) を “models/ast.dat” から読み込む。

2.3 画像生成の選択肢

OIS ソフトウェアは、大別すると 2 種類の画像生成を行うことが期待されている。ひとつは精細画像生成であり、もうひとつは簡易画像生成である。両者のおおまかな区別を表 1 に示す。ただしこれらの区別は、固定された要求仕様ではなく、訓練の用途に応じて、起動時もし

```
{
  "init_pid_file": "run/pid.txt",
  "init_port_file": "run/port.txt",
  "init_ccd_db": "data/ccd",
  "init_model_probe": "models/probe.stl",
  "init_model_asteroid": "models/ast.stl",
  "init_geological_unit_id": "models/ast.dat",
  ...
}
```

図 5: JSON の書式例 (起動用設定)

表 1: 画像生成に係る選択肢の比較

	精細画像生成	簡易画像生成
ポリゴン数	～4 億	～500 万
画素値計算	外部ライブラリ	簡易 Hapke モデル
地質情報	ポリゴン毎	一律
理学データ	オプション	無
付随処理	未定義	LIDAR 模擬 画心計算
オーバーサンプリング	必須	オプション

くはコマンドによって与えられる各種パラメータでカスタマイズされるものである。

簡易画像生成にあたっては、LIDAR[13] 模擬や、生成された画像に基づいた簡易的な画心計算等の要求が併用されることがある。精細画像生成にあたっては、理学データ等、別途提供されるデータを併用したレンダリングが要求されることがある。

画素値計算は、2 種類の手段を使い分ける。

外部ライブラリを用いた画素値計算 精細画像生成が指示された場合は、係る画像生成に必要な地質情報や角度情報 (入射角, 観測角, 位相角) を画素毎に記録した上で、起動時に読み込まれた各種属性データ等を添えて、画素値計算ライブラリを呼び出す。

簡易 Hapke モデルを用いた画素値計算 簡易画像生成のために、簡易版の Hapke パラメータを用いた反射モデル (本稿では簡易 Hapke モデルと記す) を導入する。これは精細画像生成において外部の画素値計算ライブラリが実行している計算を、地質属性やスペクトル分布等に依らず高速計算で代用するものである。

本稿で採用した簡易 Hapke モデルにおいて画素値 r を計算する式を式 1 に示す [14]。なお、この式は OIS の用途によっては変更される可能性がある。

$$r(\sigma, e, \alpha) = \frac{\omega}{4\pi \cos \sigma + \cos e} \{ [1 + B(\alpha)] p(\alpha) + H(\cos \sigma) H(\cos e) - 1 \} \quad (1)$$

ここで σ は入射角, e は観測角, α は位相角を表す. $B(\alpha)$, $p(\alpha)$, $H(x)$ は, それぞれ式 2, 式 3, 式 4 で表される.

$$B(\alpha) = \frac{B_0}{1 + \frac{1}{h} \tan \frac{\alpha}{2}} \quad (2)$$

$$p(\alpha) = \frac{1 - \xi^2}{(1 + 2\xi \cos \alpha + \xi^2)^{\frac{3}{2}}} \quad (3)$$

$$H(x) = \frac{1 + 2x}{1 + 2x\sqrt{1 - \omega}} \quad (4)$$

ここで ω , B_0 , h , ξ は, 地質によって定まる値である. OIS においては簡易的に, 前述の地質属性から代表的な値を参照する.

3 レイトレーシング

本節においては, OIS ソフトウェアの機能の要となっているレイトレーシングについて, その概略を述べる.

3.1 ソフトウェアレイトレーシングの採用

基本的な手法は, 文献 [15] の評価結果に鑑みて, 形状の計算, 影の計算共にレイトレーシングを採用することとする. これは, 特に LSS 訓練において, 陰影の描写を含めて, 高いレンダリング精度を要求されることに拠る. 精度の観点ではまた, 単精度浮動小数点演算では LSS 訓練に係る高い精度要求を満たせないため, 座標計算に係る演算は倍精度浮動小数点を用いることとした.

速度の観点では GPGPU による高速化も検討課題として挙げられるが, 本稿が対象とする画像生成装置においては, CPU 演算を採用することとした. 最大の理由は形状モデルの規模にある. 訓練に用いられる形状モデルは, 4 億ポリゴン近いモデルが存在し, 開発時点で入手できる GPU のメモリ容量では, そのようなサイズの形状モデルを収容することは困難であると判断された.

簡易画像の生成にあたっては軽量の形状モデルが用いられ, 高い精度も要求されないため, GPGPU 対応のソフトウェアを並行して開発する選択肢や, 単精度浮動小数点演算を用いる選択肢もあったが, 後述のように, CPU を用いた倍精度浮動小数点演算によるレイトレーシングでも所望の時間内に処理を終えられる目処が立ったため, 開発・保守コスト低減の観点から, 倍精度浮動小数点による CPU 演算に一本化することとした.

3.2 Voxel 分割

レイトレーシングにおいては, voxel 分割が高速化の要点のひとつとなる. 本稿においては文献 [15] に則った voxel 分割を採用し, その評価を行なった. 以下に, その voxel 分割手法の概要を示す.

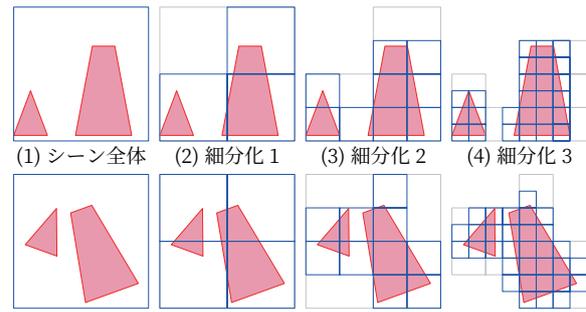


図 6: シーン毎の voxel 分割

3.2.1 Voxel 分割手法の比較

シーン毎の voxel 分割 シーン毎の voxel 分割の例を図 6 に示す. 以後, 本稿においては voxel 分割を簡易に表現するために 2 次元のシーン内で 4 分割を基本とした図を示すが, OIS ソフトウェアにおける voxel 分割は 3 次元空間内で, 各軸方向に 2 分割, 都合 8 分割することを基本として実行される.

シーン毎に voxel 分割するにあたっては, シーン内のオブジェクト配置が変化する度に voxel 分割を再実行することとなる. 本稿で想定する画像生成装置においては, 模擬する画像毎にオブジェクト配置が異なるため, シーン毎の voxel 分割を採用した場合, レンダリングの度に voxel 分割を待たねばならない.

オブジェクト毎の分割 これに対して, オブジェクト空間毎の voxel 分割の例を図 7 に示す. voxel の細分化は各オブジェクトの座標系内で実行されるため, シーン内のオブジェクト配置が変わっても, オブジェクト空間におけるオブジェクト形状が変化しない限りは, voxel 分割を再実行する必要はない.

OIS の用途に限れば, 関連する訓練を通じて, オブジェクトの変形は考慮しなくて良い. 仮に変形があったとしても, 変形のない複数のオブジェクトに予め分離することで対処可能である.

また訓練に先立っての準備時間は十分に設けることができるので, 訓練開始前に Voxel 分割を完了することで, 訓練中は voxel 分割に要する時間を実質 0 にすることができる.

レイトレーシング時の voxel 分割の影響 レイトレーシングにおいては, レンダリング時のレイの追跡に相應の時間を要する. Voxel 分割をオブジェクト単位で実施する場合, シーン内で単一のレイ (視線, 光線等) であるものを, 各オブジェクト空間毎に座標変換し, 異なるレイとして扱うこととなる. 例えば図 7 のシーンであれば, 図 8 のように 2 つのオブジェクト空間において夫々のレイを追跡することとなる. その概念図を図 9 に示す.

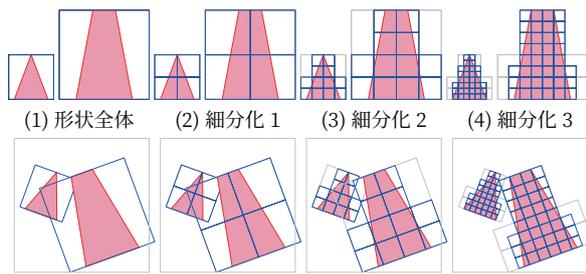


図 7: オブジェクト毎の voxel 分割

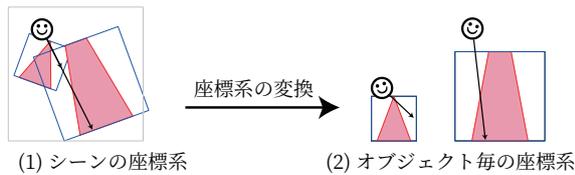


図 8: 座標系の対比

多数のオブジェクトを扱う場合は、このようなオーバーヘッドを無視できなくなる可能性がある。しかしながら OIS が扱うシーンにおいては、オブジェクト数は限られており、主要なオブジェクトは探査機と小惑星の 2 つである。訓練内容によって若干のオブジェクトが加わる程度である。

3.2.2 シーン配置の実例

実際に想定しうる小惑星と探査機の関係を図 10 に示す。同図において、左図は「はやぶさ 2」によるリュウグウの撮像例（実画像）である。右図は、この位置関係を模式的に表現したものである。リュウグウと「はやぶさ 2」のサイズは 100 倍以上の開きがあり、図中では「はやぶさ 2」は点で表現されるのみとなっている。右図の青色部分は左図を撮像したカメラ、ONC-W1[17] の視野範囲を例示している。ONC-W1 と並んで望遠側の撮像を担うカメラ、ONC-T の視野範囲も黄色で例示している。

ここで、カメラのみならず、カメラの後ろ側に位置し

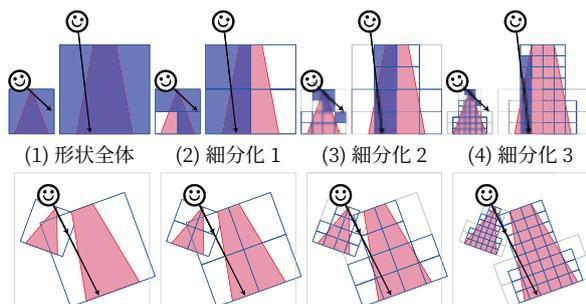


図 9: Voxel の追跡.

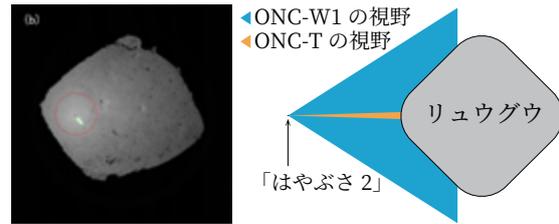
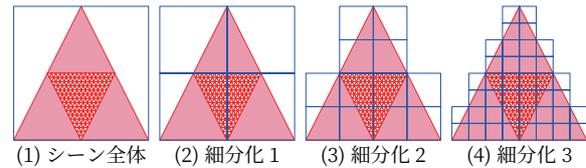
図 10: リュウグウと探査機の位置関係
左: 実画像（引用元: [16]）、右: 位置関係

図 11: ポリゴンサイズと voxel 分割の対比

ていて決して直接は描かれない探査機のオブジェクトも同時にシーン内に配置するのは、探査機が小惑星に落とす影を描写するためである。図 10 の左図にも小さな点ではあるが「はやぶさ 2」が実際に影を落としている（同図中の矢印部分）。

このように、各オブジェクトのサイズに比べて、探査機・小惑星を含むシーン全体のサイズは必ずしも小さくないことと、シーン内に配置されるオブジェクトのサイズに極端な相違があるのが、OIS における画像生成の実際である。

3.2.3 ポリゴンサイズと voxel 分割

ここでポリゴンサイズと voxel 分割の関係について検討する。

図 11 に、不均一なポリゴンサイズが混在するオブジェクトの例を示す。これはオブジェクトの一部にポリゴンサイズの極端に異なる領域が存在する例である。局所的な現象やオペレーション等を詳細にレンダリングする必要が生じた場合に、このような不均一なモデルが使われる可能性がある。

これをオブジェクト単位で均等に voxel 分割した場合、細分化が進むにつれて、サイズの大きなポリゴンに対しては過剰な voxel 分割が進行し、一方でサイズの小さなポリゴンに対しては voxel サイズが十分に細分化されないという不均衡な状態となる。このような不均衡は、大きなポリゴンで構成されるオブジェクトと小さなポリゴンで構成される局所的なオブジェクトとに分離することで対処できる可能性がある。

表 2: 開発環境の主要諸元

CPU	Intel Core i9-7900X
クロック	3.30GHz
CPU 数	1
コア数	10
PassMark 値	21997[18]
主記憶	128GB
OS	CentOS 7.5
コンパイラ	gcc 4.4.7

表 3: HIL 画像生成装置の主要諸元

CPU	Intel Xeon CPU E5-2640 v4
クロック	2.40GHz
CPU 数	2
コア数	20
PassMark 値	20765[19]
主記憶	256GB
OS	CentOS 6.8
コンパイラ	gcc 4.4.7

3.2.4 Voxel 分割の方針

以上に鑑み、本稿においては、voxel 分割を以下の方針で行う。

Voxel 分割はオブジェクト単位として事前実施することにより、訓練中に voxel 分割のコストが発生するのを抑止する。

同一オブジェクトの中にポリゴンサイズの極端な不均衡が存在する場合は、その大小や分布状況に応じて複数のオブジェクトに分離して取り扱う。

Voxel 分割は訓練の開始前に十分な準備時間をもって完了できると想定されるため、voxel 分割そのものに要する時間は本稿の検討対象外とする。

4 評価

本節においては、前節の方針に基づいた voxel 分割と、レイトレーシングの所要時間について評価を行う。

4.1 評価環境

ほとんどの性能評価には、開発環境を用いる。開発環境の主要諸元を表 2 に示す。また HIL における画像生成装置の主要諸元を表 3 に示す。PassMark 社のベンチマークで示される性能値 (Average CPU Mark) [18][19] としては、ほぼ同程度の CPU 性能となっており、用いるコンパイラのバージョンも同一とした。

なお画像生成装置の主記憶が開発環境の 2 倍となっているが、画像生成装置においては複数のタスクが並行して稼働するため、1 つの OIS プロセスが開発環境と比べて 2 倍の主記憶を占有できる訳ではない。

4.2 取り扱うデータとレンダリング条件

評価に用いた形状モデルを表 4 に示す。これらは実際の訓練に供されたリュウゴイドの形状モデル及びそれら

と同一の手法で生成された形状モデルである。均一分割モデルは、いずれも同一の仮想小惑星 (リュウゴイド) に由来しており、その相違はポリゴン分割の程度にある。すなわち、表の下位に位置するモデルは直近上位のモデルに対して、ポリゴン (三角形) を 4 分割した上で、一段階詳細な形状生成と更なるボルダー配置を施したモデルとなっている。

不均一分割モデルも、概形はリュウゴイドに準じている。名称末尾「M」のモデルは、規模の異なるポリゴンサイズが混在した形状である。この内、局所的な詳細ポリゴンを抽出したものが、末尾「S」のモデルであり、残りの大型ポリゴンで全体形状を構成するのが、末尾「L」のモデルである。また N31ML と N31MS は常に 1 組で用いられており、この組を「N31ML+S」と表現する。同様に N151ML と N151MS の組も「N151ML+S」と表現する。

各シーンにおいて入力として与えられる、主要なダイナミクス情報を表 5 に示す。ここで位置の単位は [m]、姿勢はクォータニオン (虚部、実部の順) で表現されている。

なお「F-Theta」のシーンは、F- θ 撮像系を模擬するために参考のため加えたものである。本稿で評価したカメラの中では、DCAM3 が最もこれに近い。映像出力としては、全天周映像 (ドームマスター [21]) のフォーマットで使用される形式でもある。

生成する画像は、オーバーサンプリング無しの、縦 1024 画素、横 1024 画素の画像とし、画素値計算には簡易 Hapke モデルを用いる。

係るデータに基づいてレンダリングしたシーンの例を示す。均一分割モデルのレンダリング結果を図 12 に示す。同図は 369M モデルを用いた例であるが、均一分割の各モデルにおいては、ポリゴンサイズの大小を除いて、同図と同様のレンダリング結果となる。

不均一分割モデルのレンダリング結果を図 13 に示す。同図は N31ML+S モデルを用いた例であるが、不均一分割の各モデルにおいては、ポリゴンサイズの大小を除

表 4: 形状モデルの概略

仮想小惑星, 均一分割モデル			
名称	ポリゴン		備考
	個数 [千]	サイズ	
331K	331	4m	直径 約 900m 以下同様
1.3M	1,344	2m	
5.5M	5,450	1m	
22M	22,417	50cm	
92M	92,238	25cm	
369M	368,955	13cm	
仮想小惑星, 不均一分割モデル			
N31M	30,598		混在
N31ML	22,728	50cm	全景
N31MS	7,870	8mm	詳細化部分
N151M	150,779		混在
N151ML	22,728	50cm	全景
N151MS	128,051	2mm	詳細化部分
探査機形状モデル			
hayabusa2	213	混在	約 4m×6m×3m

表 5: 各シーンの主要ダイナミクス情報等

ONC-W1 a, 不均一分割モデル	
カメラ	ONC-W1[17]
探査機位置	0, 0, 458
探査機姿勢	0.36017, 0.47590, 0.78029, 0.18692
ONC-W1 b, ONC-W2	
カメラ	「ONC-W1 b」: ONC-W1 「ONC-W2」: ONC-W2[17]
探査機位置	-194.69, -6.35, 500.27
探査機姿勢	0.79954, 0.08406, 0.27253, 0.52858
ONC-T	
カメラ	ONC-T[12]
探査機位置	-194.69, -6.35, 9000.27
探査機姿勢	0.79954, 0.08406, 0.27253, 0.52858
DCAM3, F-Theta	
カメラ	「DCAM3」: DCAM3[20] 「F-Theta」: F- θ 撮像系模擬
探査機位置	-971, 0, 528
探査機姿勢	0.57485, 0.35713, 0.65536, 0.33542

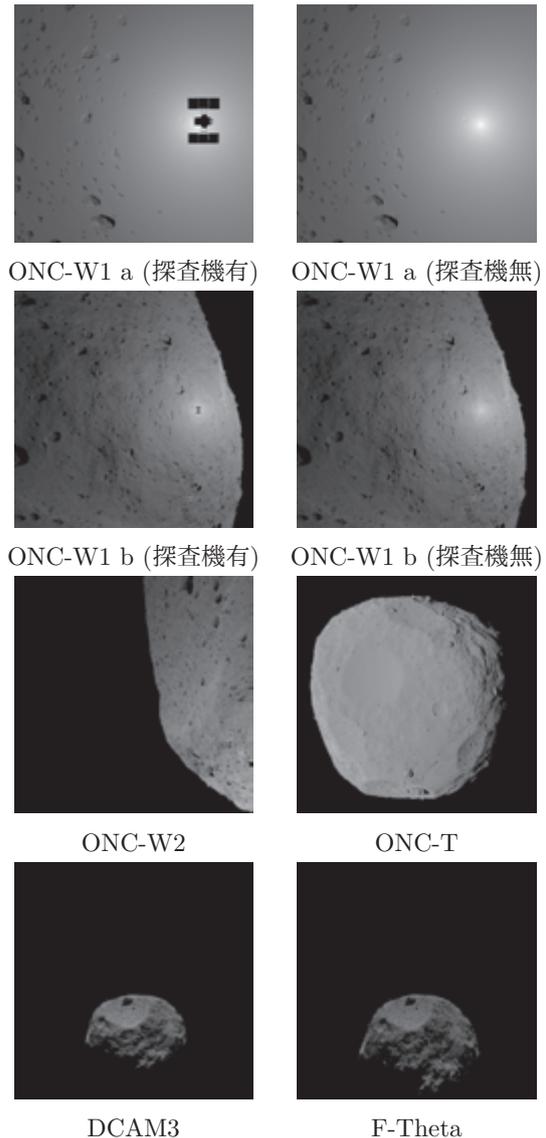


図 12: 評価に用いたシーン (均一分割モデル)

いて, 同図と同様のレンダリング結果となる. 同図中の青丸は詳細化されたポリゴン (N31MS) の範囲を示す.

4.3 Voxel 分割の評価

本節では均一分割モデルを用いて, voxel 分割の評価を行う.

4.3.1 Voxel 分割に伴う voxel 数の増加

Voxel 分割の回数と分割後の voxel 数の関係を図 14 に示す. 横軸が voxel 分割の回数, 縦軸が分割後の voxel 数を表している. 1 段階前の voxel 数との比を図 15 に示す. 横軸が voxel 分割の回数, 縦軸は voxel 分割後の voxel 数と, 一段階前の voxel 分割時の voxel 数との比を表している. これらより画像生成装置に用意される小惑

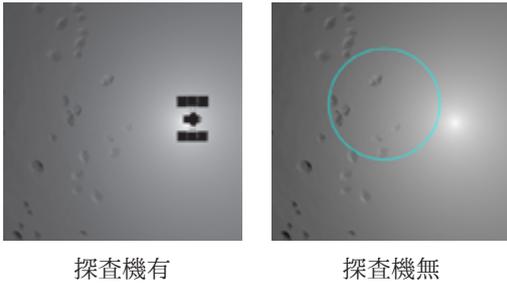


図 13: 評価に用いたシーン (不均一分割モデル)

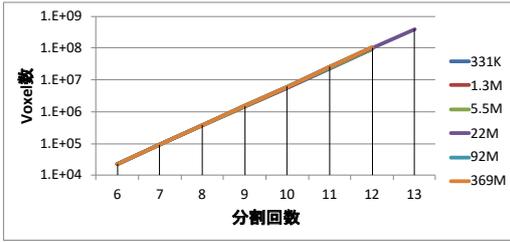


図 14: Voxel 分割の回数と分割後の voxel 数

星形状モデルについては、voxel 分割の都度、4 倍程度 voxel 数が増加していることが見て取れる。これは、それぞれの voxel 分割にあたって、所属するポリゴンを持たずに消失した voxel が半数程度存在したと考えられる。

4.3.2 キャッシュサイズ

OIS において、voxel 分割は関連する一連のレンダリングに先立って実施され、その結果がメモリ上に常駐した状態でレンダリングの指示を待つ。これらのデータは、同一形状モデルである限りにおいては再利用可能であり、不使用時はファイルシステム等にキャッシュしておくことが可能である。係るキャッシュサイズを見積もった結果を図 16, 図 17 に示す。キャッシュの内容は、各 voxel の情報と、それに属する一連のポリゴンの番号である。

横軸が voxel 分割の回数、縦軸が分割後のキャッシュサイズを表している。この図によると、主記憶容量と比較して、voxel 分割回数が 11 回を超える辺りからキャッシュサイズが急激に増加している。メモリ占有量を考え

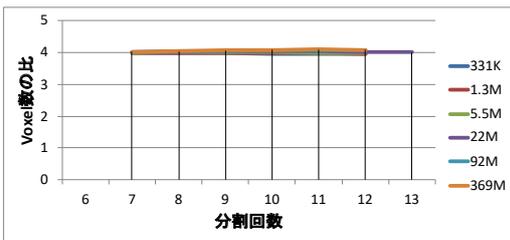


図 15: Voxel 分割後の voxel 数の比 (1 段階前の voxel 分割と比較)

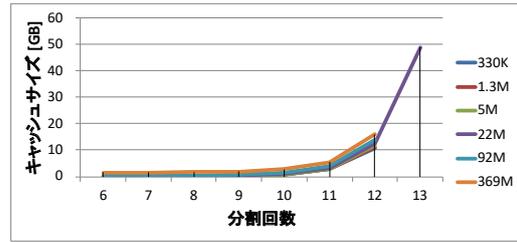


図 16: Voxel 分割の回数とキャッシュサイズ

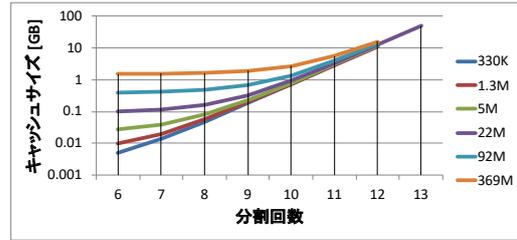


図 17: Voxel 分割の回数とキャッシュサイズ (対数スケール)

ると、画像生成装置における voxel 分割の回数は 12 回程度が現実的な上限になると考えられる。

4.3.3 計算量の見積

ここで、複数の voxel に重複して所属するポリゴンは比較的少ないと仮定してレイトラッキングに係る計算量のオーダーを見積もる。

ポリゴン数を M とすると、voxel を n 回分割した際に 1 つの voxel 内に所属するポリゴン数の平均は、前節の結果より、およそ $M/(4^n)$ となると考えられる。voxel 階層 1 回の追跡に係るコストを a とすると n 回分割した voxel の階層 ($n+1$ 階層) を追跡するコストは $a(n+1)$ のオーダーとなる。Voxel 内のポリゴン追跡に係るコストを b とすると、計算量のオーダー o は式 5 で表される。現実にはこれらの値は a や b のような単純な定数で表現できるものではないが、ここではおおまかな目安を得るために、簡略化した表現を用いる。

$$o = a(n+1) + \frac{bM}{4^n} \tag{5}$$

Voxel 分割回数 n を変化させた時に、係る計算量が最小となるのは、微分が 0 となる付近、すなわち式 6 が成立する付近と考える。

$$a + \frac{bM}{4^n} \log \frac{1}{4} = 0 \tag{6}$$

表 6: Voxel 分割回数を目安

形状モデル	分割回数を目安
331K	8
1.3M	9
5.5M	10
22M	11
92M	12
369M	13
N31M	11
N31ML	11
N31MS	10
N151M	12
N151ML	11
N151MS	12

これを変形すると、式 10 が導かれる。

$$a = \frac{bM}{4^n} \log 4 \quad (7)$$

$$4^n = \frac{bM}{a} \log 4 \quad (8)$$

$$n \log 4 = \log M + \log \left(\frac{b}{a} \log 4 \right) \quad (9)$$

$$n = \frac{\log M}{\log 4} + \frac{\log \left(\frac{b}{a} \log 4 \right)}{\log 4} \quad (10)$$

すなわち、 $(\log M)/(\log 4)$ に何らかの値を加減したものが、目安になると考えられる。本稿においては簡単のため、 $(\log M)/(\log 4)$ を voxel 分割回数を目安とみなすこととする。表 6 に、本稿で用いた形状モデルに対する voxel 分割回数を目安を示す。

以後、voxel 分割回数を“相対”値で記す場合は、各形状モデルのポリゴン数から計算した voxel 分割回数を目安に対する差を用いることとする。

ここで、実際に voxel 分割した際にポリゴンが複数の voxel にどの程度重複して含まれる状況であるかを図 18 に示す。横軸は voxel 分割回数の相対値、縦軸は各ポリゴンが所属する voxel 数の平均値を表している。1 つのポリゴンが 1 つの voxel のみに所属する場合に縦軸の値が 1 となる。現実には voxel 分割回数の相対値が -2 を超える辺りから縦軸の値が急増し、以後は指数関数的に増加している。前述の計算量のオーダの仮定は voxel 分割回数が少ない時のみ成立するものであることが、この図から読み取れる。一方で同仮定から導き出した voxel 分割回数を目安を基準に横軸を整理することで、異なるポリゴン数のモデルに対して、グラフの傾向を比較することが容易になることがわかった。

ただしこれは、広く一般的な形状モデルに対して、こ

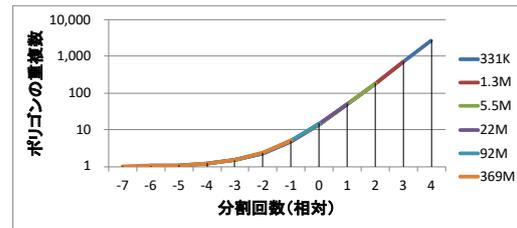


図 18: ポリゴンの重複数

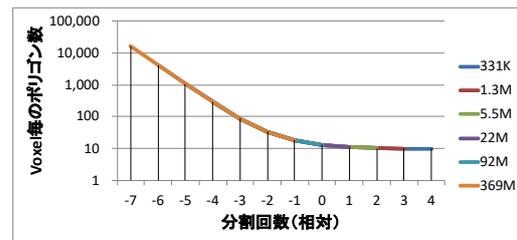


図 19: Voxel 毎のポリゴン数 (平均値)

の目安が適切であることを主張するものではない。

4.3.4 Voxel に含まれるポリゴン数

図 19 には、各 voxel に含まれるポリゴン数の平均値を示す。横軸は voxel 分割回数の相対値、縦軸は voxel 毎のポリゴン数 (平均値) を表している。Voxel 分割回数の相対値が -2 を超えた辺りから voxel 毎のポリゴン数の減少が鈍化し、相対値が 0 以上になると、voxel 毎のポリゴン数はほぼ一定となる。これはすなわち、voxel 分割回数の相対値が 0 付近を超えると、voxel 追跡のコストのみが増加し、voxel 内のポリゴン追跡に係るコストは減少しなくなる傾向にあると考えられる。

4.3.5 Voxel 分割に関するまとめ

前述の結果から、メモリ使用効率の観点では voxel 分割の回数が 12 を超えない範囲が妥当であり、また所要時間の観点では voxel 分割の相対値が 0 を超えない範囲が妥当であると考えられる。

4.4 レンダリングの所要時間に係る評価

続いて、4.2 節のデータに基づいて、各シーンのレンダリング所要時間の評価を行う。

ここで測定対象とする装置は Linux システムであり、測定以外のプロセスを完全に排除することは困難である。そのため、異常な測定値が結果に影響しないように、レンダリングするシーン毎に 30 回繰り返して所要時間を計測し、その中で所要時間の短い 25 サンプルを計測結

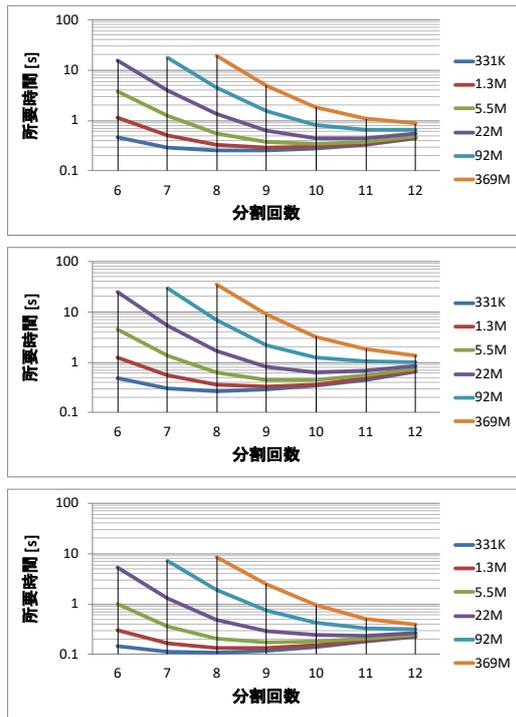


図 20: 各シーンのレンダリングに係る所要時間 (小惑星形状のみ)
上: ONC-W1 a, 中: ONC-T, 下: DCAM3

果として採用することとする。この回数は経験則によるものである。

4.4.1 均一分割モデルの個別評価 (探査機無)

小惑星形状モデルのみを用いて、均一分割モデルのレンダリングに係る所要時間を計測した結果を図 20 に示す。横軸は voxel 分割回数、縦軸は所要時間を表している。以後の結果も含めて、例として 3 種類のシーンの結果を示しているが、他のシーンについてもおおまかな傾向は同様であった。

また開発環境と画像生成装置 (HIL) との実行結果比較を図 21 に示す。比較に用いたのは“ONC-W1 a”シーンの“5.5M”モデルである。画像生成装置は開発環境と比べて他プロセスの影響が大きいため、あくまでも参考値となるが、概ね開発環境に若干勝る実行速度となっている。傾向として、画像生成装置と開発環境に決定的な相違はないと考えられるので、以後は開発環境を用いて評価を行うこととする。

図 20 の結果に基づいて、voxel 分割回数の相対値を横軸にとった比較を図 22 に示す。縦軸は、計測対象とした voxel 分割回数の中で所要時間が最短となったものを 1 とおいて相対値を記したものである。全体として、voxel 分割回数の相対値が-1 もしくは 0 の時に所要時間が最短となり、そこから離れるに従い所要時間が増加す

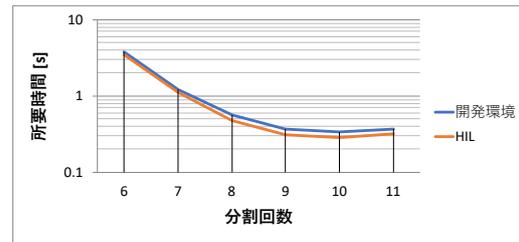


図 21: レンダリングに係る所要時間比較 (ONC-W1 a, 5.5M, 小惑星形状のみ)

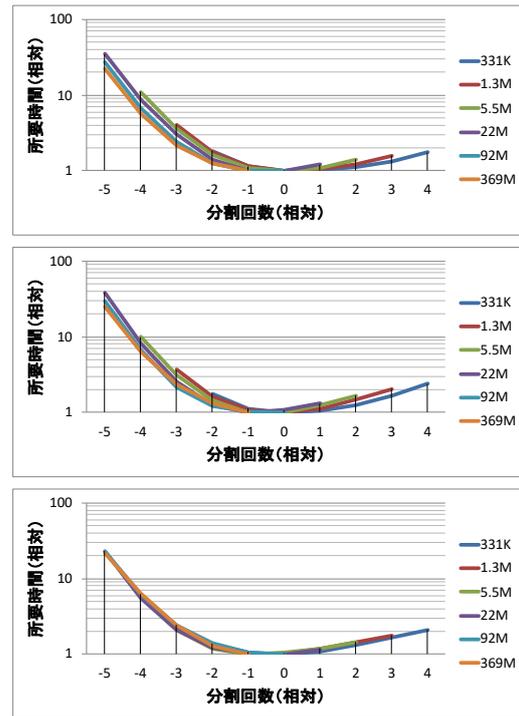


図 22: 各シーンのレンダリングに係る所要時間 (相対値, 小惑星形状のみ)
上: ONC-W1 a, 中: ONC-T, 下: DCAM3

る傾向にある。これは 4.3.5 節と矛盾しない結果である。

“369M”モデルは voxel 分割回数の相対値の上限が-1 (絶対値としては 4.3.5 節で述べた、妥当と思われる voxel 分割回数の上限である 12) であったため、voxel 分割回数の相対値が 0 以上となるまで計測している他のケースとは状況が異なるが、他のケースからの類推として、これ以上 voxel 分割回数を増やしてメモリを消費したとしても、所要時間に目立った改善は見られないと推測される。

これらの所要時間について、ばらつきを求めた結果を図 23 に示す。縦軸の標準偏差は、所要時間の平均値に対する百分率で示したものである。

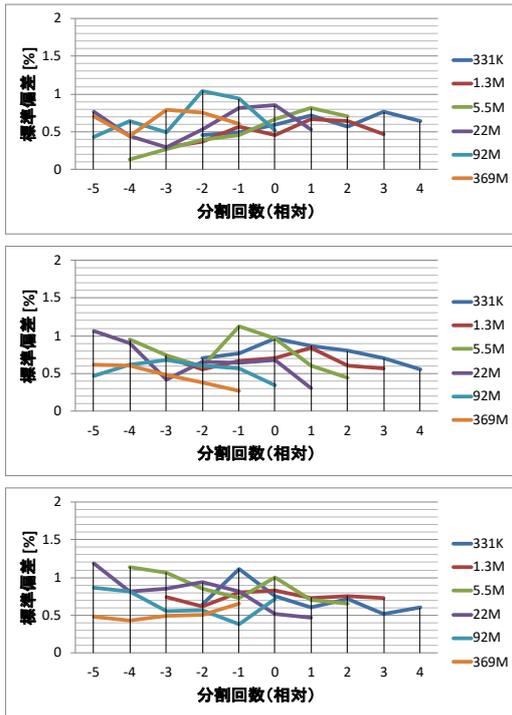


図 23: レンダリングに係る所要時間の標準偏差
(相対値, 小惑星形状のみ)
上: ONC-W1 a, 中: ONC-T, 下: DCAM3

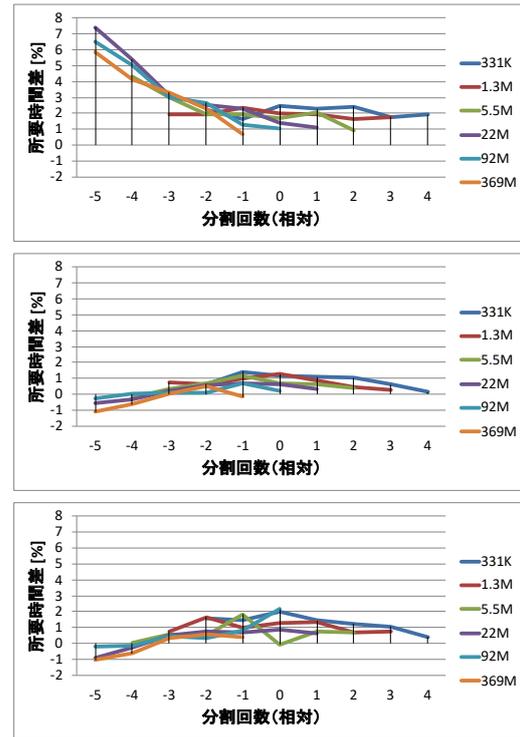


図 24: レンダリングに係る所要時間の差
(相対値, 探査機形状の有無の差)
上: ONC-W1 a, 中: ONC-T, 下: DCAM3

4.4.2 均一分割モデルの個別評価 (探査機有)

探査機の形状モデルを併用した場合と小惑星形状モデル単独の場合とで、所要時間 (平均値) の差を求めた結果を図 24 に示す。図 25 には、探査機の形状モデルを併用した場合の所要時間のばらつきを示す。図 24 における縦軸の値は、所要時間 (平均値) の差を小惑星形状モデル単独での所要時間 (平均値) に対する百分率で表したものである。所要時間が最短となることが期待される、voxel 分割回数 (相対) が-1 から 0 の付近では、その差は高々2%程度であった。これは所要時間計測の際のばらつきに対して十分大きな値ではなく、探査機の形状モデルを併用することの所要時間増は、実用上、大勢に影響ないと考えられる。

“ONC-W1 a” のケースは、voxel 分割回数が少ない領域において、所要時間が目立って増加する傾向が見られたが、所要時間が最も短くなると期待される、voxel 分割回数の相対値が-1~0 の付近では、さほどの増加傾向が見られず、これも実用上の支障は無いと判断される。

4.4.3 均一分割モデルの評価 (まとめ)

以上をまとめて、各シーンのレンダリングに係る所要時間の内、最短となるケースを図 26 に示す。図中に参考値として記載した線は、ポリゴン数の 1/4 乗の傾きを

持っている。これらの比較から、レンダリング時間の増分は、概ねポリゴン数の 1/4 乗の増加に抑えられていると考えられ、大規模形状モデルのレイトレーシングにあたっては、高々1秒余りの時間でレンダリングを終了する結果となった。

簡易画像は概ね“1.3M”から“5.5M”付近の規模の形状モデルを用いることが想定されているが、この付近のモデルは1秒に対して十分早い時間でレンダリングを実行できることを示している。この傾向は、画像生成装置を用いた測定結果図 21 から読み取れるものであり、簡易画像生成に対する要求を満たすものと考えられる。

また本稿で導入した voxel 分割回数の目安は、ポリゴン数の異なるモデルに対して適切な voxel 分割回数を導く目安として、実際の画像生成装置にも実装され、voxel 分割のパラメータ指定の簡略化に寄与している。すなわち、形状モデル毎に voxel 分割回数を指定することなく、相対的な voxel 分割回数 (-1, 0 等) を起動時のパラメータに設定するだけで、様々な規模の小惑星形状モデルに対して、概ね良好な実行結果が得られることとなった。

精細画像の生成にはオーバーサンプリングを伴うが、レイトレーシング機能単体で評価する限りにおいては、例えば4倍×4倍のオーバーサンプリングを想定しても、RIO 訓練で要求される応答時間 (10分) に対しては十

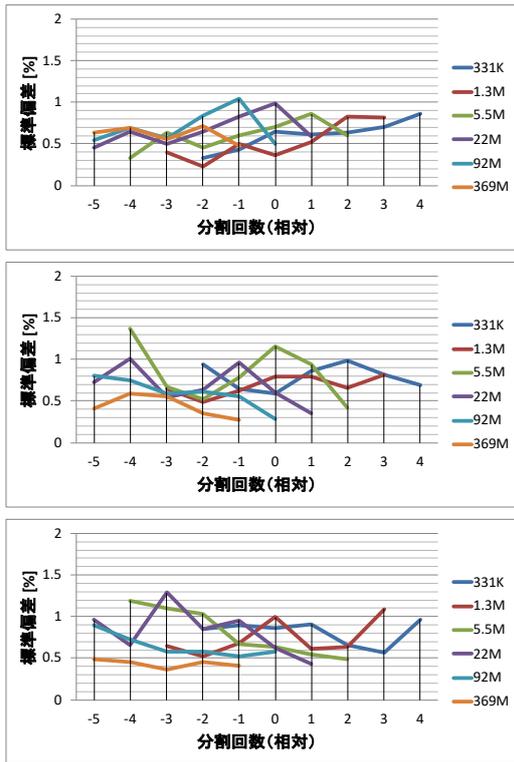


図 25: レンダリングに係る所要時間の標準偏差 (相対値, 小惑星形状+探査機形状)
上: ONC-W1 a, 中: ONC-T, 下: DCAM3

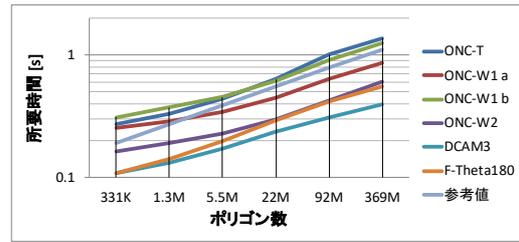
分高速にレンダリングを終了できる見込みとなった。だが実際には、外部の画素値計算ライブラリ等を含めた実行時間も鑑みつつ何倍のオーバーサンプリングまで許容されるか、HIL を稼働させる中で適切な値を推定する必要がある。

4.4.4 不均一分割モデルの評価

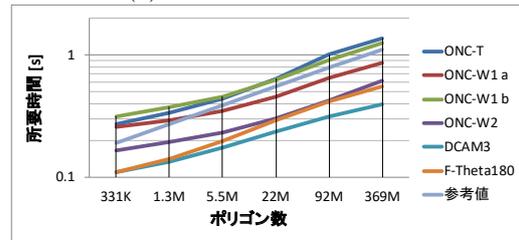
不均一分割モデルは、その用途や生成条件等によって様々な形態が想定されるが、ここでは一例を示すにとどめる。

図 27 に、N31ML+S の“ONC-W1 a”シーンにおける測定結果を示す。この図は小惑星と探査機をシーンに取めた場合の比較結果であるが、小惑星のみの場合も同様の比較結果となった。

所要時間の測定対象としては、N31M (不均一分割モデル), N31ML+S (N31M をポリゴンサイズが均一となるような 2 種類の形状モデルに分離したもの), および N31ML を構成するポリゴンとサイズが最も近い、22M を選定した。横軸が voxel 分割回数, 縦軸が所要時間を示している。ここで N31ML+S については、N31ML の voxel 分割回数を代表として示している。N31MS の voxel 分割回数は、それぞれ N31ML に対して 1 回少ない値となっている。N31ML+S においては、voxel 分割回数が



(a) 小惑星形状モデルのみ



(b) 小惑星形状モデルと探査機形状モデル

図 26: 各シーンのレンダリングに係る所要時間

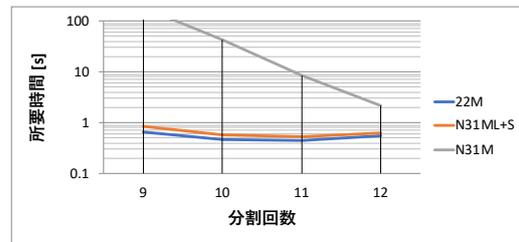


図 27: 不均一形状モデルを用いた場合の所要時間

11, すなわち表 6 より、voxel 分割回数の相対値が 0 の場合に最も所要時間が短くなっている。N31M においては、本稿で想定していた voxel 分割回数の上限である 12 回でも十分に所要時間が短くなっているとは考えられず、所要時間の短縮のためには、さらなる voxel 分割が必要と考えられる。しかしながら、これ以上の voxel 分割はメモリ利用率の観点で、現実的ではないと考えられる。

この比較結果より、不均一分割モデルをポリゴンサイズに応じた複数のモデルに分離することでポリゴンサイズの局所的な不均一を解消すれば、比較的少ないリソースでレンダリング時間の短縮が図れることが示唆された。

ただし、どのような不均一分割モデルに対してどのような効果があるかについては、更なる比較実験や考察が必要であり、均一分割モデルほどの画一的な voxel 分割の目安が見出されたものではない。

参考までに、同一シーンで N151M と N151ML+S を用いた場合の所要時間比較について、一例を表 7 に示す。この例の Voxel 分割回数 (相対) は 0 である。

表 7: 不均一形状モデルを用いた場合の所要時間

モデル	所要時間 [ms]
N151M	56,850
N151ML+S	617

5 あとがき

本稿においては、HIL シミュレータの画像生成装置におけるレイトレーシングソフトウェアの概要とその評価について述べた。

用いられる形状モデルに対して、voxel 分割回数を目安となる簡易の式を導き出すことができたため、画像生成装置におけるコンフィグレーションの構築にあたって、高速化のための試行錯誤を低減できるようになった。

開発環境における評価結果は要求要件を満足するもの

であり、それと同等以上の環境が用意された画像生成装置においても十分な性能を発揮できると期待されるものである。しかしながら画像生成装置において、複数プロセスが並行稼働する環境下にあつては、実際にコンフィグレーションを調整しながら、システム全体として所望の機能・性能を発揮できるようにするための努力が欠かせない。

一方でポリゴンサイズが不均一な形状モデルに対しては、事前にオブジェクトを分離する等の手作業が発生している。このようなケースでもオブジェクトを分離することなく、現実的なメモリ使用量でレンダリング速度を向上させることも、改善の余地として残されていると考えられる。

本稿で述べた光学機器模擬の手法は、「はやぶさ 2」ミッションに閉じず、広く関連する課題等への応用も期待されるものであり、今後も様々な要求に応じた開発が続くと考えられる。

参考文献

- [1] 佐伯孝尚, リュウグウ近傍での運用をシミュレートする— RIO 訓練, ISAS ニュース 2018 年 4 月号, 2018.4
- [2] H. Yabuta, N. Hirata, R. Honda, Y. Ishihara, K. Kitazato, M. Komatsu, A. Miura, K. Matsumoto, T. Morota, T. Nakamura, A. Nakato, T. Noguchi, T. Okada, N. Sakatani, S. Sugita, S. Tachibana, S. Tanaka, E. Tatsumi, S. Watanabe, T. Yamaguchi, Y. Yamamoto, LSS AA Team (Hayabusa2 Project), “Hayabusa2 landing site selection (LSS) training: Summary report of scientific evaluation”, 49th Lunar and Planetary Science Conference, The Woodlands, Texas, March 2018.
- [3] 三浦 昭, 山口 智宏, 本田 理恵, 横田 康弘, 千秋 博紀, 北里 宏平, 山本 幸生, 中村 智樹, 野口 高明, 本田 親寿, 山田 学, 和田 浩二, 佐伯 孝尚, 田中 智, “はやぶさ 2 着陸地点選定運用訓練のための仮想 Ryugu データ作成”, 日本惑星科学会 2017 年秋季講演会予稿集, H15, 2017.9.27-29
- [4] 渡邊誠一郎, 石原吉明, 田中智, 山口智宏, 三浦昭, 山本幸生, 平田成, 諸田智克, 坂谷尚哉, 北里宏平, 松本晃治, 藪田ひかる, “はやぶさ 2 LSS データ解析検討チーム, はやぶさ 2 着陸点選定訓練: データ解析・検討”, 日本惑星科学会 2017 年秋季講演会予稿集, H16, 2017.9.27-29
- [5] 平田成, 平田直之, 杉山貴亮, 金丸仁明, 千秋博紀, 北里宏平, 田中小百合, 西川直輝, 渡邊誠一郎, 石原吉明, 田中智, 山口智宏, 三浦昭, 山本幸生, はやぶさ 2 プロジェクト LSS データ解析検討チーム, “はやぶさ 2 着陸点選定訓練における小惑星形状復元”, 日本惑星科学会 2017 年秋季講演会予稿集, P53, 2017.9.27-29
- [6] 杉山貴亮, 平田成, 渡邊誠一郎, 石原吉明, 田中智, 山口智宏, 三浦昭, 山本幸生, LSS データ解析検討チーム, “形状復元ソフト Photoscan のはやぶさ 2 着陸点選定訓練における活用の成果と課題”, 日本惑星科学会 2017 年秋季講演会予稿集, P54, 2017.9.27-29
- [7] 坂谷 尚哉, 千秋 博紀, 荒井 武彦, 滝田 隼, 岡田 達明, 田中 智, 平田 成, 三浦 昭, 山口 智宏, はやぶさ 2 TIR チーム, 坂谷尚哉, “はやぶさ 2 着陸地点選定訓練データを用いた熱慣性決定手法の評価”, 日本惑星科学会 2017 年秋季講演会予稿集, P56, 2017.9.27-29
- [8] 諸田智克, 千秋博紀, 横田康弘, 坂谷尚哉, 巽瑛理, 杉田精司, 本田理恵, 本田親寿, 山田学, 平田成, 平田直之, 三浦昭, 山口智宏, 田中智, はやぶさ 2 光学航法カメラチーム, “はやぶさ 2 ONC データによる Ryugu 表面ラフネスの推定 着陸点選定訓練データを用いた検討”, 日本惑星科学会 2017 年秋季講演会予稿集, P57, 2017.9.27-29,

- [9] 杉田精司,*諸田智克,巽瑛理,本田理恵,本田親寿,山田学,神山徹,横田康弘,平田成,三浦昭,山口智宏,田中智,山本幸生,石原吉明,千秋博紀,“はやぶさ2 光学航法カメラチーム,はやぶさ2 ONC データの処理システムとプロダクト 着陸点選定訓練データの例”,日本惑星科学会 2017 年秋季講演会予稿集, P58, 2017.9.27-29
- [10] JAXA はやぶさ2 プロジェクト,“2018 年の小惑星リュウグウ到着にむけて 小惑星探査機「はやぶさ2」の近況”,小惑星探査機「はやぶさ2」の記者説明会 (17/12/14), 2017.12.14
- [11] Y. Takei, T. Takahashi, T. Yamaguchi, T. Saiki, A. Miura, H. Takeuchi, Y. Tsuda, “A Hardware-in-the-loop simulator for deep space touchdown operation training of Hayabusa2”, 68th International Astronautical Congress (IAC), Adelaide, Australia, September 2017.
- [12] S. Kameda, H. Suzuki, T. Takamatsu, Y. Cho, T. Yasuda, M. Yamada, H. Sawada, R. Honda, T. Morota, C. Honda, M. Sato, Y. Okumura, K. Shibasaki, S. Ikezawa, S. Sugita, “Preflight Calibration Test Results for Optical Navigation Camera Telescope (ONC-T) Onboard the Hayabusa2 Spacecraft”, Space Science Review, Volume 208, Issue 1-4, pp.17–31, July 2017.
- [13] T. Mizuno, T. Kase, T. Shiina, M. Mita, N. Namiki, H. Senshu, R. Yamada, H. Noda, H. Kunimori, N. Hirata, F. Terui, Y. Mimasu, “Development of the laser altimeter (LIDAR) for Hayabusa2”, Space Science Reviews, Volume 208, Issue 1-4, pp.33–47, July 2017.
- [14] 五味広美, 佐々修一, 山口功, 二宮哲次郎, 濱田吉郎, “Shape from shading による月平坦地の実時間検出”, 航空宇宙技術研究所報告, TR-1447, 2002.8
- [15] 三浦昭, 山本幸生, 吉川真, “「はやぶさ2」運用補助のための可視化手法 — 影の描写の検討 —”, 宇宙科学情報解析論文誌 第五号, pp.133-148, JAXA-RR-15-006, 2016 年 3 月.
- [16] はやぶさ2 プロジェクト, “探査機の影と衝効果”, <http://www.hayabusa2.jaxa.jp/topics/20180915/>, 2018 年 11 月 20 日現在.
- [17] H. Suzuki, M. Yamada, T. Kouyama, E. Tatsumi, S. Kameda, R. Honda, H. Sawada, N. Ogawa, T. Morota, C. Honda, N. Sakatani, M. Hayakawa, Y. Yokota, Y. Yamamoto, S. Sugita, “Initial inflight calibration for Hayabusa2 optical navigation camera (ONC) for science observations of asteroid Ryugu”, Icarus 300, pp.341–359, 2018
- [18] PassMark Software, “PassMark - Intel Core i9-7900X @ 3.30GHz - Price performance comparison”, <https://www.cpubenchmark.net/cpu.php?id=3035>, 2018 年 11 月 20 日現在.
- [19] PassMark Software, “PassMark - [Dual CPU] Intel Xeon E5-2640 v4 @ 2.40GHz - Price performance comparison”, <https://www.cpubenchmark.net/cpu.php?id=2752&cpuCount=2>, 2018 年 11 月 20 日現在.
- [20] K. Ishibashi, K. Shirai, K. Ogawa, K. Wada, R. Honda, M. Arakawa, N. Sakatani, Y. Ikeda, “Performance of Hayabusa2 DCAM3-D camera for short-range imaging of SCI and ejecta curtain generated from the artificial impact crater formed on asteroid 162137 Ryugu (1999 JU₃)”, Space Science Reviews, Vol. 208, Issue 1-4, pp.213–238, July 2017.
- [21] IMERSA / AFDI Dome Standards Group, “FULLDOME FILE SPECIFICATIONS”, <https://www.imersa.org/fulldome-file-specifications>, 2018 年 11 月 20 日現在.

宇宙航空研究開発機構研究開発報告 JAXA-RR-18-008
JAXA Research and Development Report

宇宙科学情報解析論文誌 第八号
Journal of Space Science Informatics Japan Volume 8

発行 国立研究開発法人 宇宙航空研究開発機構 (JAXA)
〒182-8522 東京都調布市深大寺東町7-44-1
URL: <http://www.jaxa.jp/>

発行日 平成31年3月8日
電子出版制作 松枝印刷株式会社

※本書の一部または全部を無断複写・転載・電子媒体等に加工することを禁じます。
Unauthorized copying, replication and storage digital media of the contents of this publication, text and images are strictly prohibited. All Rights Reserved.

