

2025 青山学院大学特殊講義SI(1)

リモセン × DEM - Google Earth Engine を使って DEMデータを利用してみよう！

担当：田中聰至 (フリーランスプログラマ / OSGeo JP 運営委員)

田中：フリーランス技術者 / OSGeoJP 運営委員

✉ : alt9800jp@gmail.com

最近やっていること

LiDARや機械学習による測量支援 / 屋内の地図化ツール/
点字ブロック Lineデータ作成 /Web における3D

お仕事

鳥居などの3Dアーカイブ / WebGISのコンサルティング / 水理 (野外IoT)

登壇

iOSDC2024 : <https://fortee.jp/iosdc-japan-2024/proposal/afbcd097-0da9-4073-8f48-528f007e28b7>

出没学会

FOSS4G / SotM / リモセン学会 / GISA / 地図学会 / 計画行政学会 など...

最近つくってるもの

<https://solemate-3xn.pages.dev>

<https://simplespeedtestamaranth.web.app>

知って欲しいこと：

3D地形データがどのように作られているか、それを
どのように活用するか

(ツールの「使い方」の話に終始してしまう部分があるのでご了承ください 😊)

リモートセンシングとは？

人工衛星や航空機などから地球の表面（Earth's surface）付近を観測する技術を指すことが多い

[Wikipedia : リモートセンシング](#)

Google Earth Engineとは？

大規模地理空間分析プラットフォームです。

近年ではGoogle Cloud に組み込まれ、これにより課金情報やWebAPIやコンピューティングリソースが管理されています。

衛星画像を中心に多くのデータがコレクションされており、適宜しづらりこんで呼び出すことができます。

Google Earthは閲覧用のプラットフォームだったことと比較すると、GEEは計算のためのツールと言えるでしょう。

Google Earth Engine(GEE)でできること

ブラウザ上のコードエディタを使うと、Javascriptを用いて簡易に既存データセットを呼び出すことができます。(様々組み込み関数が用意されています。)

また、近年では同様にGoogleの管理している [Google Colaboratory](#)からPythonを使ってAPI経由でリソースを取得する機会も増えてきました。

Google Earth Engineを使うと何が嬉しいか。

従来、人工衛星のデータはとても容量が大きく、手元のコンピュータ(ローカル)に膨大な容量も必要で、かつマシンスペックも強くなければいけませんでした。

しかし、Google (Cloud)が得意とするコンピューティングリソース管理により、範囲ごとに計算を切り出して並列に演算したり、広い範囲で多くの計算が必要でも、手持ちのPCの負荷を小さく解析ができるようになりました。

データカタログ

<https://developers.google.com/earth-engine/datasets?hl=ja>

90ペタバイトものデータがコレクションされている

衛星から得られたデータ(による画像データ)と解析済みのデータを分析・可視化したデータの両方を呼び出すことが可能。

衛星データの種類

光学衛星データ

衛星名	解像度(観測波長)	周期	提供元
Landsat	30m(可視光・近赤外)、100m(熱赤外)	16日	USGS/NASA
Sentinel-2	10m, 20m, 60mなど(マルチスペクトル)	5日	ESA(欧州宇宙機関)

いつのデータか、が重要

SAR衛星データ

衛星名	解像度(観測波長)	周期	提供元
Sentinel-1	10m	6-12日	ESA, EC
ALOS PALSAR	25m	5日	JAXA

前者については「コペルニクス計画」とかで検索するとロマンが見えて面白いかも。

地形データ(解析済みセット)

データ名	解像度	解析手法	提供元
SRTM (Shuttle Radar Topography Mission)	90m (高解像度化済みデータもあり:30m)	レーダー干渉測量	USGS
NASA DEM	30m	ステレオ画像から生成	NASA
ALOS World 3D	30m	ALOS衛星のPALSAR(合成開口)レーダー干渉測量	JAXA

と、このあたりの知識は詳しくは畠畠やRESTECさんによくまとまっているので参照されたし。

SARとは？？

<https://sorabatake.jp/3364/>

Sentinel-1とは？

<https://www.restec.or.jp/satellite/sentinel-1-a-1-b>

コラム：気象観測衛星「ひまわり」のデータを使いたい

JAXA ひまわりモニタ

https://www.eorc.jaxa.jp/ptree/index_j.html

ひまわりリアルタイムWeb - NICT

<https://himawari8.nict.go.jp/>

画像データ自体を入手したい場合は

気象庁のJMA ArchiveからDLしてGEEに取り込むと、GEEのデータと組み合わせて可視化ができる。

(でもすでに多分コミュニティアセットとして有志がまとめてくれてるはず)

Q. 航空写真測量とどう違うのさ? / A. By. Claudeくん

項目	航空写真測量	衛星リモートセンシング
観測高度	数百m～数km	数百km～数万km
空間解像度	数cm～数十cm(非常に高精度)	数十cm～数十m(用途により様々)
観測範囲	狭い(数km ² ～数十km ²)	広い(数千km ² ～全球)
観測頻度	低い(プロジェクトベース)	高い(数日～数週間)
コスト	単位面積あたり高コスト	単位面積あたり低コスト
天候依存	非常に高い	やや低い(光学は依存、SARは不問)
データ取得	オンデマンド(必要時に飛行)	アーカイブ+定期観測
処理の複雑さ	比較的シンプル	大気補正など複雑な処理が必要
アクセス性	契約・発注が必要	オープンデータが多い

DEMとDSMについて

DEM (Digital Elevation Model / 数値標高モデル)

DEMが欲しいパターン：地形図

DSM (Digital Surface Model / 数値表層モデル)

DSMが欲しいパターン：建物の高さ情報をしりたい時

リンク

<https://earthengine.google.com>

コードエディター

<https://code.earthengine.google.com>

視覚的に見た方が理解しやすい人はYoutubeで視聴するのも面白いかも

まずはログインからやってみましょう！

Google Earth Engineのページにアクセスして、

<https://earthengine.google.com>

The screenshot shows the Google Earth Engine homepage. At the top, there is a navigation bar with links for Platform, Datasets, Noncommercial, Commercial, Timelapse, Case Studies, FAQ, and Get Started. Below the navigation bar, a banner introduces Google Earth AI, describing it as a collection of geospatial AI models and datasets to address critical global challenges, with a link to learn more. The main feature is a large satellite map of a coastal region, showing land and water in various shades of green and blue. Overlaid on the map is the text "A planetary-scale platform for Earth science data & analysis". Below this, a smaller text states "Powered by Google's cloud infrastructure". A button labeled "Watch Video" with a play icon is visible. At the bottom left, the date "2025/11/07" is shown, and at the bottom right, the number "19".

The screenshot shows the login interface for the Earth Engine Code Editor. At the top left is a Google sign-in button labeled "Google でログイン". Below it is a placeholder icon for a user profile picture. The main text "アカウントを選択してください" (Please select an account) is displayed prominently. To the right, a user account is listed with a small profile picture, the name "S. [REDACTED]", and the email "s.[REDACTED]@gmail.com". A radio button labeled "別のアカウントを使用" (Use another account) is shown below. At the bottom, a note in Japanese asks users to review the "Privacy Policy" and "Terms of Use" before using the app. Navigation links at the very bottom include "日本語" (Japanese), "ヘルプ" (Help), "プライバシー" (Privacy), and "規約" (Terms).

Google でログイン

アカウントを選択してください

「Earth Engine Code Editor」に移動

S. [REDACTED]
s.[REDACTED]@gmail.com

別のアカウントを使用

このアプリを使用する前に、Earth Engine Code Editor の[プライバシー ポリシー](#)と[利用規約](#)をご確認ください。

日本語 ▾

ヘルプ プライバシー 規約

まずはリソースの構成を行っておくと良いです。(後述) (Google Cloudの利用だと他のサービスでも必ず行います。)

Google Cloud 2024-07-26 検索

Earth Engine / Configuration

概要 タスク 構成

構成

Cloud プロジェクトを商用または非営利として登録して、Earth Engine にアクセスします

Earth Engine は、気候、サステナビリティ、環境に関する作業で使用される地理空間分析用の優れたツールです。条件を満たしている非営利組織は無償でアクセスできます。営業活動を行う企業や行政機関は、有料の商用利用を通じて Earth Engine にアクセスできます。

商用利用の登録
料金プランを表示
続行

非営利目的で使用するための要件を満たしているか確認する
ユースケースの例を確認します。非営利目的のユーザーは、非営利目的の利用資格を定期的に確認する必要があります。
始める

2025/11/07 リリースノート

21

Earth Engine / Configuration

概要

タスク

構成

構成

Cloud プロジェクトを商用または非営利として登録して、Earth Engine にアクセスします

Earth Engine は、気候、サステナビリティ、環境に関する作業で使用される地理空間分析用の優れたツールです。条件を満たしている非営利組織は無償でアクセスできます。営業活動を行う企業や行政機関は、有料の商用利用を通じて Earth Engine にアクセスできます。

商用利用の登録

 [料金プラン](#) を表示

[続行](#)

非営利目的で使用するための要件を満たしているか確認する

 [ユースケースの例](#) を確認します。非営利目的のユーザーは、非営利目的の利用資格を定期的に確認する必要があります。

[始める](#)

会社または企業（B コーポレーションを含む）

公立または私立の教育機関（教職員、学生を含む）

非営利団体

政府、行政機関

ニュース メディア、ジャーナリスト

Earth Engine のトレーナーまたはトレーニー

その他

✓ 組織の種類を選択する

✓ 非営利目的での利用条件を確認する

✓ プランの選択

④ 担当業務の説明

Earth Engine で取り組んでいる内容は、次のいずれかのカテゴリに該当しますか？

緩和策

例: 温室効果ガス排出量 / CO₂ 換算値の削減または回避

適応

例: 気候変動の影響への適応を人々やコミュニティが行えるように支援する

保護と保全

例: 生物多様性と生態系を保全するための陸地と海洋を対象とした介入

次のいずれかの用途で Earth Engine を使用する予定はありますか？ *

農業、大気質 / 大気汚染、建築 / 都市設計、コンサルティング、教... ▼

次へ

⑤ Review summary

必要な API の有効化

❶ 登録を完了するには、Earth Engine API を有効にしてください。

[Google Earth Engine API](#) ⓘ

無効

フィードバックを送信

キャンセル

有効にする

非営利目的での使用が登録されました

×

概要ページで Earth Engine API にアクセスし、データセットを調べて分析を始めましょう。

続行

EECU 時間を管理する



使用量（EECU 時間）の 1 日の上限を作成、編集、表示します。

割り当てを管理

Cloud プロジェクトは非営利の目的で登録されています



プロジェクトが非営利の資格要件を満たさなくなった場合は、登録情報を変更するか、商用に更新してください。

登録の管理

 Earth Engine / 概要

 概要

 タスク

 構成

Earth Engine へようこそ

Earth Engine は、地球規模の地理空間分析を可能にする優れたツールであり、気候、サステナビリティ、環境科学、環境への影響の研究に 10 年以上にわたって使用されています。地理空間データセットを分析および可視化できます。Earth Engine Data Catalog の 80 ペタバイトを超えるデータを利用することも、独自のデータを使用することもできます。



エクスペリエンスの拡張



接続

[コードエディタ](#)、[Colab](#)、または任意のクライアントを使用して [Earth Engine API](#) にアクセスします



探す

一般公開されているアーカイブでは、50 年以上にわたり蓄積してきた過去のデータセットや科学的データセットなど、日々更新されるさまざまな [データセット](#) をご利用になれます



分析

Google の Data Catalog からのデータの取り込み、アセットのインポート、ML モデルの適用、[BigQuery へのエクスポート](#)などの機能を利用して、地理空間分析を行うことができます

 リリースノート

◀

- いただいた回答からすると、非営利目的での Earth Engine の使用が可能です。
- 非営利団体登録の場合、お支払いプランは不要です。
- Earth Engine の構成ページで、1 日の使用量上限（EECU 時間）を作成、編集します。

EECU (Earth Engine Compute Unit) の割り当て
(これら辺で請求書情報の割り当てが必要かも)

プロジェクトがない場合はまずは新しいプロジェクトをつくる

The screenshot shows the 'Create New Project' dialog box in the Google Cloud interface. At the top, there is a navigation bar with the Google Cloud logo, a search bar containing 'スラッシュ (/) を使用してリソース、ドキュメント、プロダクトなどを検索', a 'Search' button, and icons for settings, notifications, help, and more.

The main area has a heading '新しいプロジェクト' (New Project). Below it, a yellow callout box contains a warning message: '割り当て内の残りのプロジェクト数は 17 projects 件です。プロジェクトの増加をリクエストするか、プロジェクトを削除してください。詳細' (The number of remaining projects in the allocation is 17 projects. Request project increase or delete the project). A 'Manage Quotas' button is also present in this box.

The 'Project Name*' input field is highlighted with a blue border and contains the text 'earthengine-project-2025-11-07'. To the right of the input field is a help icon (a question mark inside a circle). Below the input field, a note says 'プロジェクト ID: eighth-network-477523-k6 後で変更することはできません' (Project ID: eighth-network-477523-k6 cannot be changed later).

The 'Edit' link is located just below the note.

The 'Location*' dropdown menu is set to '組織なし' (No organization). To its right is a '参照' (Reference) button. Below the dropdown, a note says '親組織またはフォルダ' (Parent organization or folder).

At the bottom of the dialog, there are two buttons: '作成' (Create, in blue) and 'キャンセル' (Cancel).

実際にコードエディターを触ってみましょう！

<https://code.earthengine.google.com/>

The screenshot shows the Earth Engine code editor interface. At the top, there's a navigation bar with tabs for Scripts, Docs, Assets, and a New button. Below the navigation bar is a search bar labeled "Filter scripts...". The main workspace is titled "New Script *" and contains the following code:

```
i 1 console.log("Hello GEE!")
2
```

To the right of the code editor is a panel with tabs for Inspector, Console, and Tasks. The Console tab is active, displaying the output "Hello GEE!" in JSON format. Below the code editor and the panel is a large map of North America and parts of Mexico and the Caribbean. The map shows state/province boundaries and major cities. Labels on the map include:

- バンクーバー Vancouver
- ワシントン州 Washington
- モンタナ州 Montana
- ノース・ダコタ州 North Dakota
- ミネソタ州 Minnesota
- サウス・ダコタ州 South Dakota
- ワイオミング州 Wyoming
- オレゴン州 Oregon
- アイダホ州 Idaho
- ネバダ州 Nevada
- ユタ州 Utah
- コロラド州 Colorado
- カンサス州 Kansas
- ミズーリ州 Missouri
- ネブラスカ州 Nebraska
- アイオワ州 Iowa
- アイオイー州 Iowa
- インディアナ州 Indiana
- ケンタッキー州 Kentucky
- ウェスト・バージニア州 West Virginia
- ワシントンD.C. Washington
- サンフランシスコ San Francisco
- ラスベガス Las Vegas
- 口サンゼルス Los Angeles
- メキシコ州 Mexico
- グアダラハラ Guadalajara
- メキシコシティ Ciudad de México
- ハバナ La Habana
- マイアミ Miami
- ヒューストン Houston
- ダラス Dallas
- テキサス州 Texas
- ルイジアナ州 Louisiana
- アラバマ州 Alabama
- ジョージア州 Georgia
- フロリダ州 Florida
- ハバナ La Habana
- キュー・バ Cuba
- サンクト・ドミニゴ Santo Domingo
- プエルトリコ Puerto Rico
- ノース・カロライナ州 North Carolina
- ミシシッピ州 Mississippi
- アーカンソー州 Arkansas
- オクラホマ州 Oklahoma
- テネシー州 Tennessee
- ノース・カロライナ州 North Carolina
- サウス・カロライナ州 South Carolina
- アラバマ州 Alabama
- ジョージア州 Georgia
- フロリダ州 Florida
- マイアミ Miami
- ハバナ La Habana
- キュー・バ Cuba
- サンクト・ドミニゴ Santo Domingo
- プエルトリコ Puerto Rico

The map also includes labels for the Great Lakes and the Atlantic coast. In the bottom left corner of the map area, there's a timestamp: "2025/11/03 10:42". On the far right, there are buttons for "地図" (Map) and "航空写真" (Aerial Photo). The top right corner of the map area has a "北大西洋" (North Atlantic) label.

コードエディタとしてこのようにJavaScriptの文法が使える。

```
console.log("Hello GEE!")
```

```
var a=1  
var b=2  
var c=a+b  
print(c)
```

相模原市淵野辺5丁目付近のデータを見てみよう

35.566958, 139.401946

Google Mapsのデータとの比較もしてみる

隨時「Reset」ボタンを押すといい感じ。

事例(ハンズオン)

青山学院大学周辺の浸水リスクを可視化してみよう！



やること

Step1: 地形データを読み込んで可視化

Step2: 標高を小分けにして抽出(低地判断)

Step3: 水域データを読み込む

Step4: 水域からの距離を分析する

Step5: 水捌けの良さを分析する

Step6: リスクのスコア化

(Step7: 計算結果を数値で表示する)

Step1 地形データを読み込んで可視化してみよう！！

```
// 対象地域の設定(相模原市周辺)
var center = ee.Geometry.Point([139.401946, 35.566958]);
var roi = center.buffer(15000).bounds();

// 地図の中心を設定
Map.centerObject(center, 11);

// DEMデータの読み込み
var dem = ee.ImageCollection('JAXA/ALOS/AW3D30/V3_2')
  .select('DSM') // 地表面の標高を選択
  .mosaic()       // 複数タイルを結合
  .clip(roi);    // 対象地域で切り取り

// 標高を色分けして表示
Map.addLayer(dem,
  {min: 0, max: 100, palette: ['blue', 'green', 'yellow', 'orange', 'red']},
  'Elevation (m)');

// 標高の最小値・最大値を確認
print('標高の統計:', dem.reduceRegion({
  reducer: ee.Reducer.minMax(),
  geometry: roi,
  scale: 30,
  maxPixels: 1e9
}));
```

Step2 標高を小分けにして抽出してみよう！(低地かどうかを判断したい！)

```
// ステップ1のコードを実行した後に追加

// 標高5m未満のエリアを抽出
var lowLand5m = dem.lt(5).selfMask();

// 標高5m～10mのエリアを抽出
var lowLand10m = dem.gte(5).and(dem.lt(10)).selfMask();

// 標高10m～20mのエリアを抽出
var lowLand20m = dem.gte(10).and(dem.lt(20)).selfMask();

// 低地エリアを色分けして表示
Map.addLayer(lowLand20m, {palette: ['yellow']}, 'Elevation < 20m');
Map.addLayer(lowLand10m, {palette: ['orange']}, 'Elevation < 10m');
Map.addLayer(lowLand5m, {palette: ['red']}, 'Elevation < 5m');

// 標高5m未満の面積を計算
var area5m = lowLand5m.multiply(ee.Image.pixelArea()).reduceRegion({
  reducer: ee.Reducer.sum(),
  geometry: roi,
  scale: 30,
  maxPixels: 1e9
});

print('標高5m未満の面積 (km2)：', ee.Number(area5m.get('DSM')).divide(1000000));
```

Step3 水域のデータを読み込んでみよう！

```
// ステップ1-2のコードを実行した後に追加  
  
// 水域データの読み込み(過去36年間の水の存在頻度)  
var water = ee.Image('JRC/GSW1_4/GlobalSurfaceWater')  
  .select('occurrence')  
  .clip(roi);  
  
// 50%以上の頻度で水が存在する場所を抽出(恒常的な河川・湖)  
var permanentWater = water.gte(50).selfMask();  
  
// 水域を青色で表示  
Map.addLayer(permanentWater, {palette: ['blue']}, 'Rivers and Water Bodies');
```

JRC Global Surface Water: 人工衛星で観測した水域データ

これ自体もSAR衛星からのデータや光学データからの抽出がされているところがミソ。

Step4 水域からの距離分析を行う

```
// ステップ3のコードを実行した後に追加  
  
// 水域からの距離を計算(ピクセル単位)  
var waterDistancePixels = permanentWater.fastDistanceTransform().sqrt();  
  
// ピクセル単位をメートルに変換(1ピクセル=30m)  
var waterDistance = waterDistancePixels.multiply(30);  
  
// 距離を色分けして表示(近い=青、遠い=白)  
Map.addLayer(waterDistance,  
  {min: 0, max: 1000, palette: ['darkblue', 'blue', 'lightblue', 'white']},  
  'Distance from Water (m)');  
  
// 河川から100m以内のエリアを抽出  
var near100m = waterDistance.lt(100).selfMask();  
Map.addLayer(near100m, {palette: ['purple']}, 'Within 100m from River');
```

Step5 水捌けの良さを定量的に示したい！

斜面の角度をDSM(DEM)データから分析してどれくらい水が流れやすそうか見積もる。

```
// ステップ1のDEMデータを使用  
  
// 傾斜角度を計算(度単位)  
var slope = ee.Terrain.slope(dem);  
  
// 傾斜を色分けして表示(平坦=緑、急=赤)  
Map.addLayer(slope,  
  {min: 0, max: 30, palette: ['green', 'yellow', 'red']},  
  'Slope (degrees)');  
  
// 平坦地(傾斜3度未満)を抽出  
var flatArea = slope.lt(3).selfMask();  
Map.addLayer(flatArea, {palette: ['cyan']}, 'Flat Areas (slope < 3°)');  
  
// 平坦地の統計  
print('傾斜の統計:', slope.reduceRegion({  
  reducer: ee.Reducer.minMax(),  
  geometry: roi,  
  scale: 30,
```

Step 6-1 評価のためのスコアリング：標高の低さによるリスク

```
// ステップ1-5のコードを実行した後に追加

// 【要因1】 標高リスク(0-3点)
var elevationRisk = ee.Image(0)
  .where(dem.lt(5), 3) // 5m未満: 3点
  .where(dem.gte(5).and(dem.lt(10)), 2) // 5-10m: 2点
  .where(dem.gte(10).and(dem.lt(20)), 1); // 10-20m: 1点

// 標高リスクを表示
Map.addLayer(elevationRisk.updateMask(elevationRisk.gt(0)),
  {min: 1, max: 3, palette: ['yellow', 'orange', 'red']},
  'Elevation Risk');

print('標高リスクの分布を確認してください');
```

Step 6-2 評価のためのスコアリング：河川からの距離によるリスク

```
// 【要因2】 河川距離リスク(0-3点)
var distanceRisk = ee.Image(0)
  .where(waterDistance.lt(100), 3)           // 100m未満: 3点
  .where(waterDistance.gte(100).and(waterDistance.lt(300)), 2) // 100-300m: 2点
  .where(waterDistance.gte(300).and(waterDistance.lt(500)), 1); // 300-500m: 1点

// 河川距離リスクを表示
Map.addLayer(distanceRisk.updateMask(distanceRisk.gt(0)),
  {min: 1, max: 3, palette: ['lightblue', 'blue', 'darkblue']},
  'Distance Risk');

print('河川距離リスクの分布を確認してください');
```

Step 6-3 評価のためのスコアリング：平坦地であることのリスク

```
// 【要因3】 傾斜リスク(0-1点)
var slopeRisk = ee.Image(0).where(slope.lt(3), 1);

// 傾斜リスクを表示
Map.addLayer(slopeRisk.updateMask(slopeRisk.gt(0)),
  {palette: ['purple']},
  'Slope Risk (Flat Areas)');

print('平坦地(水が溜まりやすいエリア)を確認してください');
```

Step 6-4 リスク計算(スコアリング)

```
// 3つのリスクを合計(最大7点)
var totalRisk = elevationRisk.add(distanceRisk).add(slopeRisk);

// 総合リスクを可視化
Map.addLayer(totalRisk.updateMask(totalRisk.gt(0)),
  {min: 1, max: 7, palette: ['lightgreen', 'yellow', 'orange', 'red', 'darkred', 'purple']},
  'Total Flood Risk');

// リスクレベルの説明
print('=====');
print('総合リスク評価:');
print('1-2点: 低リスク (light green - yellow)');
print('3-4点: 中リスク (orange)');
print('5-7点: 高リスク (red - purple)');
print('=====');
```

Step 7 コンソールに計算結果を表示させる

```
// ステップ6までのコードを実行した後に追加  
  
// リスクレベルごとの面積を計算  
var riskLevels = [1, 2, 3, 4, 5, 6, 7];  
  
riskLevels.forEach(function(level) {  
  var riskArea = totalRisk.eq(level).multiply(ee.Image.pixelArea());  
  var area = riskArea.reduceRegion({  
    reducer: ee.Reducer.sum(),  
    geometry: roi,  
    scale: 30,  
    maxPixels: 1e9  
});  
  
  print('リスクレベル ' + level + ' の面積 (km2):',  
    ee.Number(area.get('constant')).divide(1000000));  
});
```

Extra : 画像として出力したい

```
// =====  
// 追加: エクスポート用の設定(オプション)  
// =====  
  
// 結果を画像としてエクスポート  
Export.image.toDrive({  
  image: totalRisk.visualize({min: 1, max: 7, palette: ['lightgreen', 'yellow', 'orange', 'red', 'darkred', 'purple']}),  
  description: 'FloodRisk_Fujisawa_Tamagawa',  
  scale: 30,  
  region: roi,  
  maxPixels: 1e9  
});
```

先にGoogle Driveを保存先に指定する必要がある。

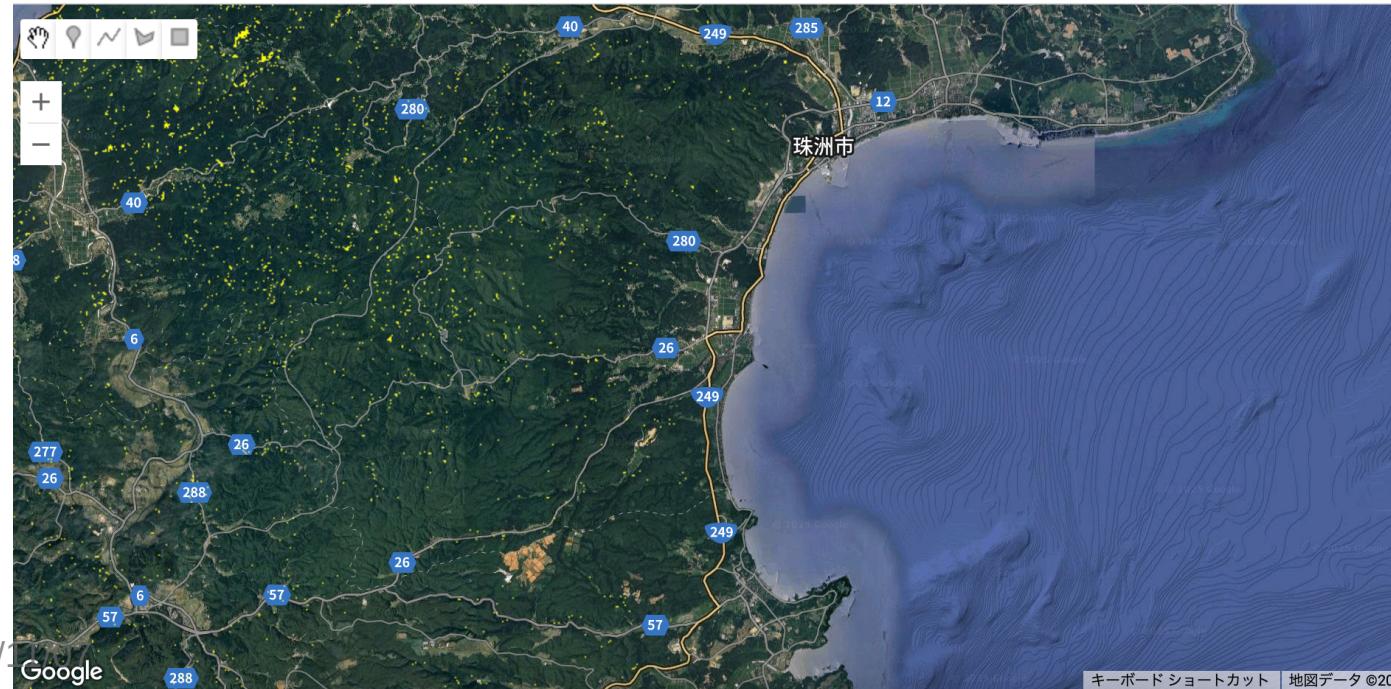
能登半島地震(2024年1月1日)前後で起きた地滑りを見てみよう

[ポイント]

DEMデータが存在しないような場所ではどのような解析ができるか？

人工衛星から得られたデータをどのように解析するか？

(こちらは講義時間の進み方次第で取り扱います。)



GEEではSARのデータを使った地表面の変化を確認することができ、この値の違いを時期ごとに比較することにより、表面の様子の差分を抽出することができます。

地滑り前 = 植生がある → SAR反射が強く取得できる
地滑り後 = 裸地になる → SAR反射が弱くなる

というような原理。

(ちなみに、InSARのデータを用いると位相差を確認できるので、どれくらい地面が隆起したかなどを確認することもできます。だいち二号のお仕事：

https://www.gsi.go.jp/uchusokuchi/20240101noto_insar.html)

県道26号線周辺の地滑りの様子がスタジオダックビルさんと東京大学の渡邊研究室によってCesium Ionをつかってストーリーテリングにされています。

<https://ion.cesium.com/stories/viewer/?id=a4bbf02c-dd2e-4a16-9556-6543ace0b96d#slide-id-186298>

鵜飼漁港の位置

<https://maps.app.goo.gl/d7GQ2M7PbVZfSPnR9>

<https://www.openstreetmap.org/#map=17/37.401226/137.242917>

Sentinel-1 SAR画像の比較

使用データ:

- 地震前: 2023年12月1日～12月31日の平均
- 地震後: 2024年1月1日～1月31日の平均
- センサー: Sentinel-1(VV偏波、Descending軌道)

変化量の計算と地滑り抽出

計算式:

変化量 = 地震後 - 地震前

地滑り検出の条件:

- ① SAR後方散乱が-2.5dB以上減少
→ 地表面が粗くなった(土砂崩れ)
- ② 傾斜角が20度以上
→ 地滑りが起きやすい急斜面
- ③ 標高50m以上
→ 海域を除外

レイヤー構成:

1. 変化量マップ(赤白青)

- 赤: SAR減少(地滑りの可能性)
- 青: SAR増加

2. 地滑り候補地(黄色)

- 条件を満たした地点

3. 分析範囲(黒枠)

- 能登半島北部

Step1 地形データの読み込み

```
// ----- 設定 -----
var area = ee.Geometry.Rectangle([136.8, 37.2, 137.4, 37.6]);
Map.centerObject(area, 10);

// ----- データ取得 -----
var sar = ee.ImageCollection('COPERNICUS/S1_GRD')
  .filterBounds(area)
  .filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
  .select('VV');

// 地震前: 2023年12月
var before = sar.filterDate('2023-12-01', '2024-01-01').median();

// 地震後: 2024年1月
var after = sar.filterDate('2024-01-01', '2024-01-31').median();

// 地形データ
var dem = ee.Image('USGS/SRTMGL1_003');
var slope = ee.Terrain.slope(dem);
```

Step2 変化量計算

```
// SAR後方散乱の変化量(dB)
var change = after.subtract(before);

// 地滑り条件
// 1. SAR後方散乱が大きく減少(-2.5dB以下)
// 2. 急傾斜地(20度以上)
// 3. 陸地(標高50m以上)
var landslide = change.lt(-2.5)
  .and(slope.gt(20))
  .and(dem.gt(50));
```

Step3 変化量を可視化する

```
// 変化量マップ
Map.addLayer(change.updateMask(dem.gt(0)),
  {min: -4, max: 4, palette: ['red', 'white', 'blue']},
  '変化量(赤=減少、青=増加)');

// 地滑り候補地
Map.addLayer(landslide.selfMask(),
  {palette: ['yellow']},
  '地滑り候補地');

// 分析範囲
Map.addLayer(ee.Image().paint(area, 1, 2),
  {palette: 'black'},
  '分析範囲');
```

Google Earth Engineで扱う画像のライセンスについて

以下のデータはパブリックドメインとして提供されているようです。

- Landsat: NASAは米国連邦機関のため著作権なし、自由に利用可能
- MODIS: 同様にNASAデータ
- Sentinel: ESAのオープンデータポリシー、商用利用含めて自由

DEMデータや気象データにはCCライセンスで提供されているものが多くあります。

(これらのデータの提供先となるプラットフォームはGoogle Earth Engineだけではなかったりする)

データセットカタログの [Terms of Use](#) セクションを確認すると各データセットについて詳しく書いてあります。

作成したデータについては研究や教育目的での利用が許可されています。

背景地図にGoogle Mapsが含まれる場合も別途地図の著作権について留意する必要があります。

© 2025 Google

データ出典: Copernicus DEM (ESA)

のような書き方をすると良いです。

他にどんなデータがあるか？

土壤水分量データ (Soil Moisture Active Passive)

観測手法 : Lバンドマイクロ波放射計

データID : NASA_USDA/HSL/SMAP10KM_soil_moisture

解像度 : 約 9 km

観測深度 : 表層 0-5 cm

観測タイミング : 日次

NDVI (正規化植生指標)

植物は近赤外が強く、赤色が弱く反射されることを利用してデータを抽出する。

Night Light

夜間の観測で、地球から確認できる光をデータ化する。

類似サービス

Tellus

<https://www.tellusxdp.com/>

日本発の衛星データプラットフォーム
ブラウザ上でデータ検索・解析が可能で
Jupyter Notebookベースのエディターを持っています。
QGISからの読み込み用のAPIも提供されています。
だいち2号(ALOS-2) SAR画像やSentinel-1/2など

NASA Worldview(NASA AppEEARS)

<https://worldview.earthdata.nasa.gov>

インターラクティブな地図ビューアで、24時間以内のデータの閲覧などもできます。

VEGA

Google Earth Engine Python APIを利用してデータを取得しやすくしているサービス
(By RESTEC)

<https://rs-training.jp/square/vega/>

<https://x.com/emmyeil/status/1986632418682667373>

なんか本が出るらしいぞ

Try もっと3Dで見てみたい！！

地形データなどを3D TilesやRGB Terrainのような形式にしてみよう！

人工衛星のデータを使ったチャレンジもおすすめ！！

<https://solafune.com/ja>

<https://www.spaceappschallenge.org>

今月は可視化強化月間なのでXやInstagramで探してみましょう！

<https://30daymapchallenge.com>